AD-A087 190 LOGICON INC SAN DIEGO CA TACTICAL AND TRAINING SYSTE--ETC F/G 17/9
GROUND CONTROLLED APPROACH CONTROLLER TRAINING SYSTEM (GCA-CTS)--ETC(U)
JUN 80 G BARBER, J BOLLENBACHER, D BREWTON N61339-77-C-0162

UNCLASSIFIED NAVTRAEQUIPC-77-C-0162-3 NL

1 OF 8
AD
A087190

LEVEL III A061036 (12)

REPORT NAVTRAEQUIPCEN 77-C-0162-3

GROUND CONTROLLED APPROACH CONTROLLER TRAINING SYSTEM SYSTEM DOCUMENTATION

GENE BARBER, JOHN BOLLENBACHER, DAN BREWTON,
DAVID HARRY, MARY HICKLIN, CATHERINE MEYN, GAIL SLEMON
LOGICON, INC.
TACTICAL AND TRAINING SYSTEMS DIVISION
POST OFFICE BOX 80158
SAN DIEGO, CALIFORNIA 92138

June 1980

FINAL REPORT FOR PERIOD 23 SEPTEMBER 1977 – 23 FEBRUARY 1980

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: NAVTRAEQUIPCEN 77-C-0162-3	2. GOVT ACCESSION NO.: AD-A087 290
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): Ground Controlled Approach Controller Training System (GCA-CTS) System Documentation	5. TYPE OF REPORT & PERIOD COVERED: Final Report, 23 SEP [redacted]77 - 23 FEB [redacted]80
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Gene Barber, John Bollenbacher, Dan Brewton, David Harry, Mary Hicklin, Catherine Meyn, Gail Slemon	8. CONTRACT OR GRANT NUMBER(s): N61339-77-C-0162
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Logicon, Inc., Tactical & Training Systems Div. P. O. Box 80158 San Diego, California 92138	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 7796-1P1
11. CONTROLLING OFFICE NAME AND ADDRESS: Naval Training Equipment Center Orlando, Florida 32813	12. REPORT DATE: June 1980
	13. NUMBER OF PAGES: 748
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Adaptive Training
Air Traffic Control (ATC)
Digital Speech Recognition
Ground Controlled Approach Controller Training System (GCA-CTS)
Performance Measurement (continued)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the hardware and software which satisfy the functional requirements for an automated adaptive training system for the training of Ground Controlled Approach (GCA) controllers. Hardware descriptions include the system controller, the trainee station and the instructor station. Software descriptions include modes of operation, speech understanding, speech generation, performance measurement, and the simulation of pilot, environment, radar and displays.

19. KEY WORDS (Continued)

Precision Approach Radar (PAR)
Speech Recognition
Speech Synthesis
Speech Understanding
Voice Data Collection

FOREWORD

The Ground Controlled Approach Controller Training System is a unique training system insofar as it provides automated adaptive training for a primarily verbal task. As an experimental prototype, it represents the first attempt to apply isolated phrase recognition technology within a computer based instructional (CBI) framework. Use of speech recognition in this application allows the features of CBI to be married with the benefits of an environmental simulator. The GCA-CTS provides both automated instruction and graded practice, using adaptively selected problems in a simulated air traffic control environment.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (CONT)

LIST OF ILLUSTRATIONS (CONT)

LIST OF ILLUSTRATIONS (CONT)

LIST OF TABLES

LIST OF TABLES (CONT)

LIST OF TABLES (CONT)

SECTION I

INTRODUCTION

BACKGROUND

The Ground Controlled Approach Controller Training System, the GCA-CTS, provides basic training in the conduct of ground controlled approaches using simulated precision approach radar (PAR) equipment. The PAR indicator provides aircraft elevation, azimuth and range information on final approach. The PAR controller's task commences when he or she assumes responsibility for the control of an aircraft after handoff from the pattern controller. That responsibility terminates when the aircraft reaches decision height, although the controller continues to give transmissions until the aircraft passes landing threshold or executes a missed approach.

During the approach the controller issues course corrections and glidepath messages over a voice channel to enable the pilot to effect a safe approach even during periods of low ceiling and visibility, regardless of the NAVAID receiving equipment in the aircraft. A well-defined, precise radio terminology (R/T) serves as the vehicle for this communication. The controller training problem therefore involves teaching the student to interpret the radar display, to determine appropriate corrections and transmissions to communicate this information to the pilot in a standard format, and, in addition, to coordinate with other air traffic control (ATC) personnel.

THE GCA-CTS

The GCA-CTS is an experimental prototype training system (as contrasted with a training device) designed to provide basic training in GCA procedures. It is designed to ensure that competent trainees master the basic skills within the five-day time frame allowed in the present course. Because of the many features of the GCA-CTS, some students complete basic training in less than five days, and enrichment exercises are provided. The GCA-CTS provides automated, individualized instruction with objective performance assessment and numerous instructional aids including illustrated texts, computer-aided instruction, adaptive problem selection, detailed performance summaries, and annotated replays.

The GCA-CTS benefits the students in other ways as well. It relieves them of pseudo-pilot duties which do not contribute to the acquisition of controller skills but which must be performed with the current training device. It also provides the faster students with opportunities for the acquisition of advanced skills, since post-graduate training is available for those students who complete the basic course quickly. Another major advantage of the system is that it relieves the instructor of many of those routine duties which encroach on his or her training management time.

The previous laboratory GCA-CTS demonstrated the feasibility of a GCA controller training system in which the student's verbal behavior is automatically monitored and scored with the aid of commercially available speech

recognition hardware. In addition, it demonstrated that a syllabus could be constructed and that automated adaptive training of the task with objective performance measurement was possible.

The experimental prototype GCA-CTS embodies all the lessons learned in the laboratory system and incorporates additional sophisticated training techniques. It is designed for motivated and responsible students. It is a system that can provide a challenging and interesting learning environment for the individual student. This requires a course adaptively tailored to meet individual needs, with clearly defined objectives which are challenging but attainable.

FOCUS OF THE SYSTEM DOCUMENTATION

The Training/Functional Design Report, delivered in February of 1978, described the behavioral objectives which the student must attain in order to pass the training course. It then described the course syllabus which was designed to meet the behavioral objectives. Finally, it detailed the functional requirements of a system which could support the training envisioned for the experimental prototype GCA-CTS.

The present document describes the hardware and software which satisfy these functional requirements. The report consists of five sections and a set of appendixes. Section II describes the hardware environment. Section III details the design of the special purpose devices required by the GCA-CTS. Section IV covers the software environment which will support the applications routines. Section V discusses the design of these applications routines.

Brief descriptions of each program in GCA-CTS are included in Appendix A. Common variable definitions and file structures are described in Appendixes B and C, respectively. Appendixes D and E are the compile and load macros for the system. Appendixes F, G, and H include the cross-references for time and range scheduled routines, interprocessor identifications and common variables. Appendix I is a glossary of Aircraft/Pilot/Environmental (APE) Simulation local variables. Appendix J presents elevation and azimuth zone interpretation. Appendix K describes the runtime stack allocation in both processors. Appendix L gives the load on call cross-reference tables. Appendix M shows the error explanations offered to the trainee.

SECTION II

THE HARDWARE ENVIRONMENT

OVERVIEW

The operational hardware is combined into three assemblies. Each assembly will normally be at a separate location. The main assembly is the system controller, configured as a double-bay cabinet 46 inches wide by 32 inches deep by 70 inches high. A second assembly is the trainee station, consisting of a desk holding several computer peripheral devices. The third assembly is the instructor station, also a desk holding several peripheral devices. The stations may be located up to 100 cable feet from the system controller in different directions. The system controller contains two central processing units, disk and diskette storage, and audio input and output units. The stations provide audio, visual, and manual interfaces to facilitate training and instructor monitoring.

Figure 1 presents a hardware block diagram. Figure 2 indicates equipment grouping and cabling.

Most of the equipment is available commercially. Logicon, however, has designed four items. Particulars of each of the three equipment groups are presented in the following paragraphs.

THE SYSTEM CONTROLLER

This is contained in a double-bay cabinet, Data General Corporation (DGC) Model 1012L. This cabinet and most of its contents are Government-furnished equipment (GFE) and, as such, should need no lengthy description. These GFE items include:

2 each – Eclipse S/130 CPU with various options and a connecting interprocessor bus.

1 each – 10 Megabyte Disk Storage Unit, accessible from either CPU.

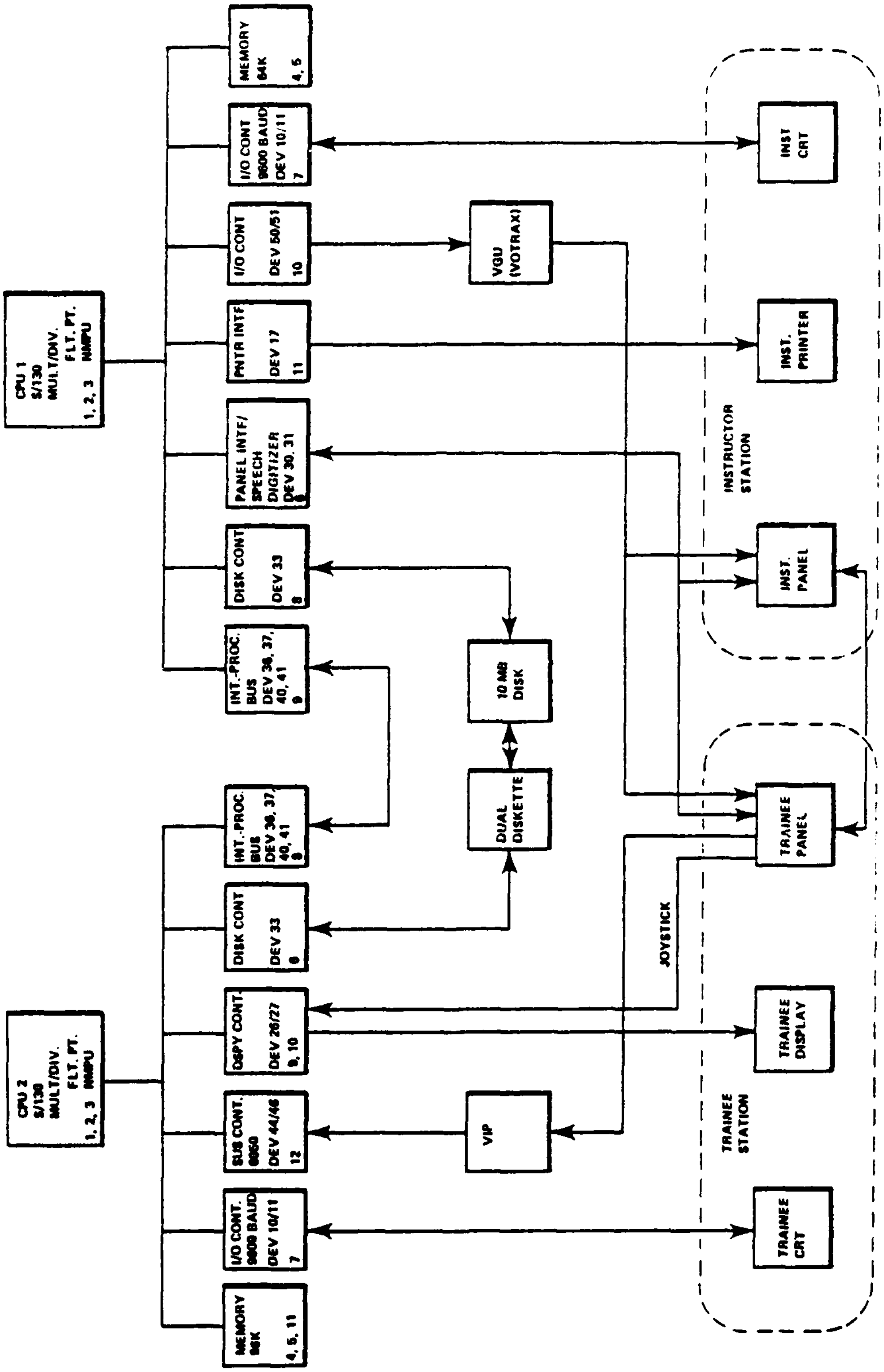
1 each – Dual Diskette Unit, also accessible from either CPU.

In addition to the above GFE items, the system controller also contains:

1 each – VS-6.4 Voice Generation Unit made by the Votrax Division of the Federal Screw Works.

1 each – Threshold 500 Voice Input Preprocessor made by Threshold Technology, Inc.

1 each – Special interface card made by Logicon; contained within CPU 1.



NOTES:
1. NUMBERS IN LOWER LEFT CORNERS ARE SLOT LOCATIONS IN CPU CABINET
2. CPU 2 SLOT 11 HAS TERMINATOR ONLY
CPU 1 SLOT 12 HAS TERMINATOR ONLY.

Figure 1. GCA-CTS Hardware Block Diagram

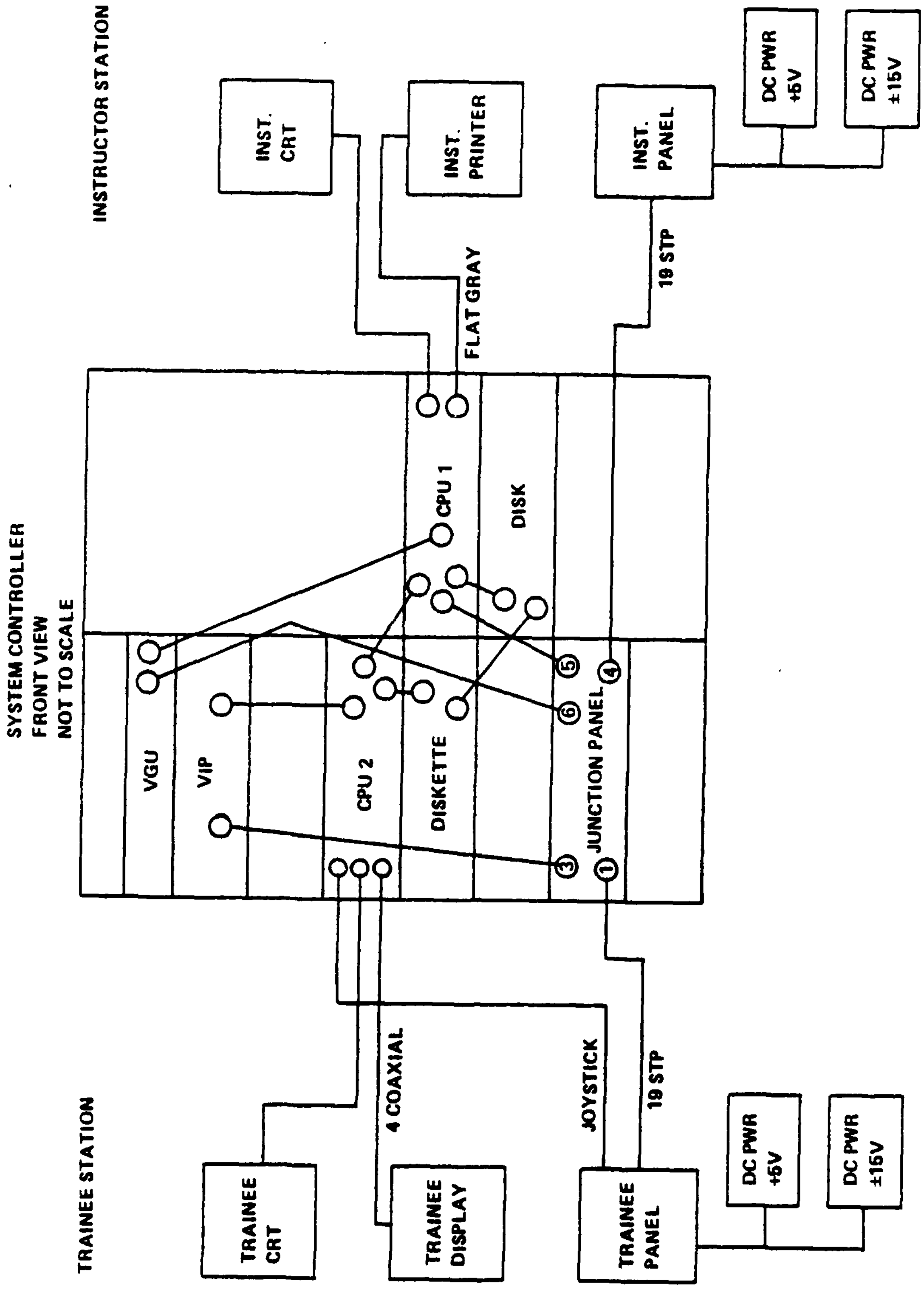


Figure 2. GCA-CTS Major Cabling

1 each - Junction Panel made by Logicon to simplify cabling and maintenance.

These items are described below.

THE VOICE GENERATION UNIT (VGU). This is a rack-mounted unit, 3.5 inches in height. When provided with properly coded data at the rate of about 15 bytes per second, this unit will generate easily understood voice signals suitable for driving a loudspeaker.

Panel controls allow adjustment of the speech rate, pitch, and level. In this system the VGU can provide the voice of the pilot, the pattern controller, or a PAR controller.

The unit is normally repaired by card replacement; six cards are involved.

THE VOICE INPUT PREPROCESSOR (VIP). This is a rack-mounted unit, seven inches in height. It is slide mounted, and its two printed circuit assemblies are easily accessible when it is extended on the slides.

This unit accepts a balanced audio input signal at a nominal 2.5 volt RMS level. When this input is present (from the trainee), the unit continuously generates 32 bits of information available on a back panel connector. These constitute two 16-bit words which may be sampled by a CPU. In this system they are sampled about 450 times per second by CPU 2. The CPU makes an assessment, based on this input and various internal tables, as to the word or phrase spoken.

The unit will normally be repaired by returning one or both circuit boards to the factory for repair. Loaner cards are available from the manufacturer.

LOGICON INTERFACES. This is an electronic assembly constructed as an interface card for CPU 1. It responds to device codes 30 and 31. It performs two functions:

a. It acts as a link between the lights, switches, and alarms in the trainee and instructor stations.

b. It provides encoding and decoding of audio data to 16-bit words and controls the data channel storing and accessing of those data. This, in conjunction with the digital disk storage, allows audio recording with instant random access for replay.

This unit is discussed further in Section III.

THE JUNCTION PANEL. This is discussed in Section III.

THE TRAI''EE STATION

The trainee station consists of a desk which holds three major components:

a. A Megatek MG552 graphic CRT display

b. A Data General Corporation Model 6053 Video Display Terminal with keyboard (GFE)

c. A Trainee Panel designed by Logicon to provide lights, buttons, sounds, and a joystick for a trainee interface

Below the desk surface, near the rear, is a power distribution strip and two DC power supplies for the Trainee Panel.

THE GRAPHIC DISPLAY. The purpose of this display is to present graphic, simulated-radar images to the student. It presents a 21-inch display with a resolution of 4096 on either axis. It is controlled by four video signals from a vector generator card within CPU 2 in the system controller.

THE VIDEO DISPLAY TERMINAL. This unit operates at 9600 baud and functions as the normal console input/output device for CPU 2.

THE TRAINEE PANEL. This unit is designed by Logicon to simulate actual equipment to provide realism to the training. It is positioned to the right of the radar-simulating display. Outside dimensions are 17 inches wide by 12 inches deep by 11 inches high. The front panel slopes backwards at 15 degrees from the vertical.

The unit contains Logicon-designed circuitry plus circuit cards for the Megatek joystick and the Threshold Technology VIP preamplifier. Panel lights, switches, and audible alarm are programmed as device 30 of CPU 1.

Audio circuitry allows microphone input and headset or speaker output for a variety of audio sources/destinations including the instructor station, the VGU, the VIP, and the device 31 recording/random playback function.

Details are presented in Section III.

DC power for this unit comes from supplies mounted below the desk top on the backside of a modesty panel.

THE INSTRUCTOR STATION

The instructor station is generally similar to the trainee station; two of the major equipments, however, are different. The major equipments are:

a. A Data General Corporation Model 6053 Video Display Terminal with keyboard as used in the trainee station (GFE)

b. A Tally Model 1602 serial character printer

c. An Instructor Panel designed by Logicon to permit audio communication with the trainee and exercise monitoring

Below the rear desk surface is a power distribution strip and two DC power supplies for the instructor panel.

THE SERIAL PRINTER. This printer is Tally Model T1602. It quietly prints 160 characters per second, bidirectionally on an original and up to four carbon copies. Up to 132 characters may be printed per line.

The printer is controlled by bytes, transmitted on eight parallel lines from a controller in CPU 1.

The printer is used operationally to provide trainee performance evaluations, diagnostic messages, summary reports, etc.

THE INSTRUCTOR PANEL. This unit functions as an intercom to the student. It also allows audible monitoring of the various voice sources in the system. It is designed by Logicon and is housed in an enclosure 17 inches wide by 12 inches deep and 7 inches high.

Details are presented in Section III.

SECTION III

SPECIAL HARDWARE

OVERVIEW

Logicon has developed a subsystem involving several audio, visual, and switch elements to facilitate communications between trainee, instructor, and computer. There are four assemblies involved and each is discussed in a separate paragraph below.

Functions of this subsystem are as follows:

a. To provide an audio intercom between trainee and instructor

b. To provide a method of recording several minutes of trainee voice with playback of any portion with an access time of one second maximum

c. To allow trainee and/or instructor to hear the above playback or the VGU output

d. To allow computer sensing of certain switch positions at the instructor and trainee stations

e. To allow computer control of certain lights and an audio alarm at the instructor or trainee station

f. To provide indication to the computer of the completion of a trainee's voice input and its approximate level

g. To provide indication to the computer of the completion of VGU audio output

This subsystem was designed and implemented with due regard to reliability, maintainability and other ILS factors.

PANEL INTERFACE/SPEECH DIGITIZER

This unit is constructed on an interface card which is installed in a slot of CPU 1. The CPU communicates with it as device 30 and 31. The unit is connected via a 100 foot cable to the trainee panel.

Device 30 acts as an interface to allow the CPU to control lights and an audio alarm at the trainee station and a light at the instructor station. It also allows the sensing of switch positions and audio levels.

Device 31 performs the general function of an audio recorder/playback unit for the trainee's voice. It makes use of a special integrated circuit called a Continuously Variable Slope Delta Modulator (CVSD). This unit is made by Harris Semiconductor as device HC-55516. It will either encode or decode an audio input. The encoded result is 16,000 bits per second. These

form 1,000 computer words per second to be stored in CPU memory buffers by data channel action. This CPU in turn must store the buffers in real time on the ten megabyte disk. On playback, the process is reversed with the CPU having the option of selecting any desired disk sectors for playback.

All analog and digital signals between this unit and the trainee panel are sent differentially on shielded, twisted pairs of wires within the cable. To reduce the number of wires involved, the device 30 I/O to and from the trainee station is transmitted/received as 32-bit serial sequences.

PANEL HARDWARE DESCRIPTION. Figure 3 presents a block diagram of the device 30 circuitry. The reader is assumed to be familiar with hardware interfacing of Data General Corporation computers.

At the top of Figure 3 are two shift register sections, each of 16-bits capacity. A is loaded first by a computer DOA command. Then B is loaded. Following the DOB, the 32 bits are automatically stepped out in serial fashion-to control lights on trainee and instructor panels and an audible alarm.

At the center of Figure 3 are two shift register sections totaling 32-bits. These are loaded by serial switch data from the trainee panel. Upon reception of 32 bits a done flip-flop is set and an interrupt is generated. The computer can then input the data by a DIA and a DIB command. Normally, data are received only when changes occur. However, the computer can at any time send a start command which sets a busy flip-flop and initiates a 32-bit transmission from the trainee panel.

PANEL PROGRAMMING. Communication with the panel is by means of standard I/O instructions. The effect of these instructions is described below.

Output Command Functions. Lights, alarm, etc. are controlled by two 16-bit words output from the training system controller by a DOA command followed by a DOB command. Both commands must be used even though a single bit is changed. Bits indicate desired conditions (1 = on, 0 = off). The commands will become effective at the trainee panel approximately 30 microseconds after the DOB command is given. Busy need not be set. Bit significance is given in Table 1. All controlled functions are on the trainee panel except DOB, bits 2 and 3. A steady amber light occurs if both amber and amber flashing bits are set. Also, amber and green lights can be on in the same switch if both bits are set.

Start and Clear Functions. Start sets the busy flip-flop and causes the trainee panel to send 32 bits to the interface. These bits will be sent within 40 microseconds. When they have been received, busy will be cleared and done will be set. This will cause an interrupt unless interrupt disable bit 3 is effective as the result of a MSKO command. The interrupt routine should clear the done flip-flop by a clear command or an I/O reset. The student's panel will automatically send a stream of 32 bits without a start command when certain events occur at the panel. These likewise will set a done flip-flop as discussed above. Events which will initiate the 32-bit transfer are a change in any of the monitored functions, DIA bits.

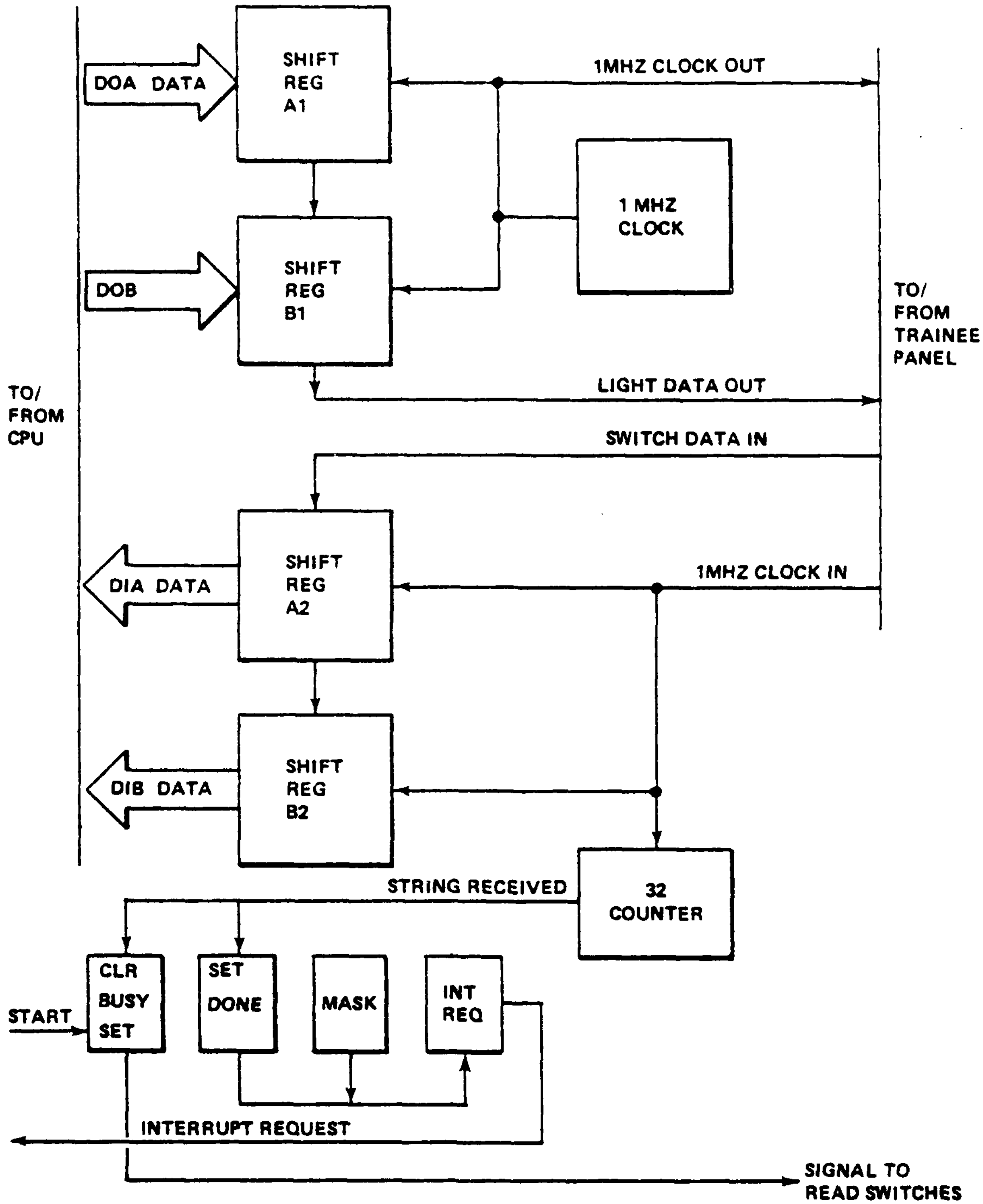


Figure 3. Device 30 Block Diagram

TABLE 1. SIGNIFICANCE OF BITS IN DEVICE 30 OUTPUT COMMANDS

DOA

Bit	Button	Condition	
0	3	Amber	
1	3	Amber Flashing	
2	5	Amber	
3	5	Amber Flashing	
4	7	Amber	
5	7	Amber Flashing	
6	SUPER	Amber	
7	SUPER	Amber Flashing	
8	270.8	Amber	Frequency
9	270.8	Green	Select
10	318.8	Amber	Frequency
11	318.8	Green	Select
12	Alarm, Audible		
13	270.8	Amber	Frequency
14	318.8	Amber	Monitor
15	REQUEST	White	

DOB

Bit	Button	Condition	
0	CLEARED	Green	
1	W/O	Red	
2	ICS	Amber	Instructor
3	ICS	Amber Flashing	Station
4	Unused		
5	Unused		
6	Unused		
7	Unused		
8	Unused		
9	Mike key status*		
10	$VRO status*		
11	Student voice status*		
12	Unused		
13	Wave Off button status*		
14	Unused		
15	Unused		

*These status bits have no effect upon the device and are set only for use by the replay and performance measurement software.

Input Request Functions. These commands are to be used only to read the 32 bits received when the done flip-flop is set. These commands should be used in tandem, the last one may contain the clear function. The meaning of these 32 bits is shown in Table 2. A change in either direction of DIA bits 0 through 13 causes a transmission sequence. Changes in DIB bits do not initiate a transmission.

TABLE 2. SIGNIFICANCE OF BITS IN DEVICE 30 INPUT COMMANDS

DIA

Bit	Meaning
0	3 depressed
1	5 depressed
2	7 depressed
3	SUPER depressed
4	270.8 depressed
5	318.8 depressed
6	270.8 monitor depressed
7	318.8 monitor depressed
8	REQUEST depressed (momentary button)
9	FOOT SWITCH depressed
10	VOTRAX active within past 0.5 second
11	Student voice active within past 0.2 - 0.4 second
12	Instructor's ICS button depressed
13	W/O depressed (momentary button)
14	Unused
15	Unused

DIB

Bit	Meaning
0–3	Student voice level*
4 . . . 15	1234_8, a constant bit pattern used for validity checking

*Voice level is a four bit representation of peak student voice level since last reset. Bit 0 is most significant bit. Auto reset occurs about 0.1 second after DIA bit 11 falls. The count ranges from 3 to 13 and approximately correlates to steady meter readings as follows:

Count	Meter Reading
7	0.1
10	0.2
11	0.4
13	0.8 or more

DIGITIZER HARDWARE DESCRIPTION. Figure 4 presents a block diagram and timing for the device. Buffers A and B are in the S/130 memory and can be of 256 to 2048 words in length. For the sake of illustration, the timing diagram was drawn to shown disk timing for the 256 word buffers.

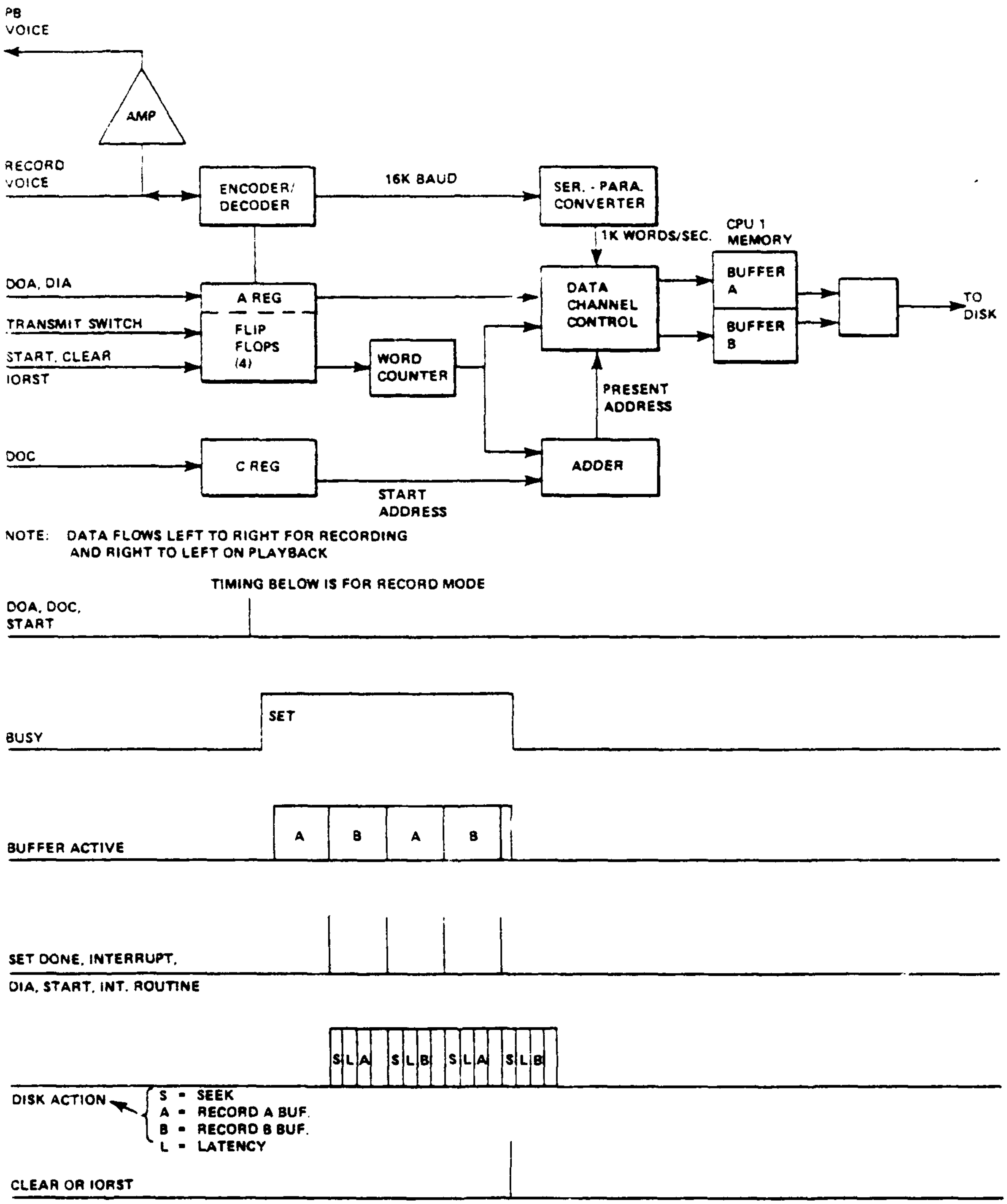


Figure 4. Device 31 Block Diagram and Timing

DIGITIZER OPERATION. In the description following, 1024 word buffers are assumed. Longer buffers require more memory but save disk seek and latency time. This buffer length is considered to be optimal for the prototype and is wired into the unit, however it is changeable. To initialize the device for record and playback, the CPU must issue a DOC command to indicate the starting logical address in the data channel map of the first buffer to be used.

Recording. To place the unit in operation, a DOAS command should be issued, ensuring that bit 0 of the output word is 0 (for "record"). This sets the busy flip-flop, clears the done flip-flop, and enables the unit.

The unit starts filling the first buffer via the data channel. A buffer will fill in approximately one second. When 1024 words have been stored, the done flip-flop is set causing an interrupt. Busy remains set. The interrupt service routine determines which buffer was filled, clears the done flip-flop by an NIOS and initiates a disk store operation of that buffer. In the meantime, the other buffer has been selected for filling with voice data. The second buffer will have a starting address 1024 words greater than that of the first buffer. Approximately every second this action will repeat, alternately filling the buffers.

There should be no need to change the contents of the C register which contains the start address of buffer A. If it is to be changed, it is recommended that it be changed by the interrupt routine. Buffer B locations must follow buffer A locations.

A buffer-filling process will continue until the clearing of busy.

The program must keep track of where on the disk the data are stored. Note that interrupts occur after the data are acquired. The time of acquisition as determined by the computer clock at interrupt time is, accordingly, the time of completion of the one-second block.

As long as busy remains set with the A register-bit 0 cleared, the process is repeated. When busy is cleared, recording stops immediately. Busy can be cleared by an I/O reset or an NIOC. This action does not set done or cause an interrupt. The final buffer may be only partly full when the NIOC is issued. It may be stored on disk if desired.

Playback. To playback the recorded voice, bit 0 of the device's A register must be set and the busy flip-flop must be set. The unit will then start reading out the contents of buffer A (which must previously have been filled from the disk). This starting buffer address is contained in the device's C register.

When buffer A has been read out, the done flip-flop will be set and an interrupt issued. Readout will continue from buffer B and then continue toggling between A and B every second setting done at the end of each buffer.

This action will continue until the busy flip-flop is cleared. When busy is cleared (I/O reset or NIOC) action will stop immediately.

Commands. Action of various commands and functions is shown in Table 3.

TABLE 3. MEANING OF DEVICE 31 COMMANDS

Command	Function
Start	Sets busy and clears done. Busy must be active for the unit to record or playback. Start is used to initiate operation or to clear done when it is desired to continue operation.
Clear	Clears busy and done. Device will stop immediately. It does not affect the A or C register.
IORST	Same as clear, but it also clears the A register (see below). This causes recording at the start of the A buffer when busy is set.
DOA	The A register controls the mode and buffer selections: Bit 0, 0 = record mode, 1 = playback mode Bit 1, 0 = Buffer A, 1 = Buffer B
DIA	Reads bits 0 and 1 of device A register Bit 0 is as set by DOA to indicate present mode Bit 1 is as follows: 0 = unit is selecting buffer A 1 = unit is selecting buffer B
DOC	Used to output starting address of buffer A. Buffer B starts 1024 words higher. This command should not be issued when in the midst of a buffer operation. It can be changed at the completion of a buffer.

Note: The device contains a word counter which indicates the relative position in a buffer that the next data channel access (record or playback) is to use. It is incremented after each access. When it reaches 1024, done is set, the word counter is cleared and the other buffer is made active.

The word counter is also cleared by either:

- I/O Reset (which also sets record mode at start of buffer A)
- DOA 31 (used to set either record or playback mode and the initial buffer)

The clear command does not reset the word counter.

Effects on the System. While recording or playing back, the unit will require approximately 1024 memory accesses per second. The associated disk action will require another 1024 accesses.

The unit will issue one interrupt per second and the disk will presumably issue two interrupts per second (at the end of each seek and each sector read/write). The maximum seek and latency time is 95 milliseconds.

Disk capacity required is 2000 bytes per second of voice. To store speech for a typical eight-minute exercise requires 960,000 bytes or 17 percent of the capacity of one of the removable cartridge disks used in the system.

THE TRAINEE PANEL

The use of this panel was discussed in the Functional Design Report.

This unit is part of the trainee station. It was designed by Logicon. It contains Logicon-designed circuitry along with a speech preamplifier circuit from Threshold Technology and a joystick digitizing circuit from Megatek. All trainee related controls and indicators are located on the front panel except for the foot switch.

Figure 5 is a view of the front panel. Table 4 describes each of the elements on the front panel.

The trainee panel is connected to the system controller by two 100-foot cables entering at the rear.

All power for the trainee panel is provided by two external DC power supplies through a single rear-mounted connector.

Figure 6 is a simplified block diagram. In the upper left a serial stream of 32 bits and a separate clock are received. In about 30 microseconds these data are stepped into a shift register. The parallel output of the shift register determines the state of the lights and the audio alarm. Transients occurring during the shift register loading are too brief to interfere with the lights. One light is actually on the instructor panel and is controlled by a differential logic signal to that panel.

In the lower left is the transmission circuitry for panel switches, VGU active, trainee voice active and voice level. These data are parallel-loaded into a 32-bit shift register and stepped out in serial fashion to device 30 in CPU1. The data are also recirculated in the shift register.

Transmission is initiated by receipt of a BUSY level. It also occurs automatically whenever there is a change in any of the inputs to shift register section A. This is accomplished by a comparator which is interrogated 20 times per second to determine any difference in present switch positions from those previously transmitted and held in the shift register.

Other blocks to the right in the figure are self-explanatory.

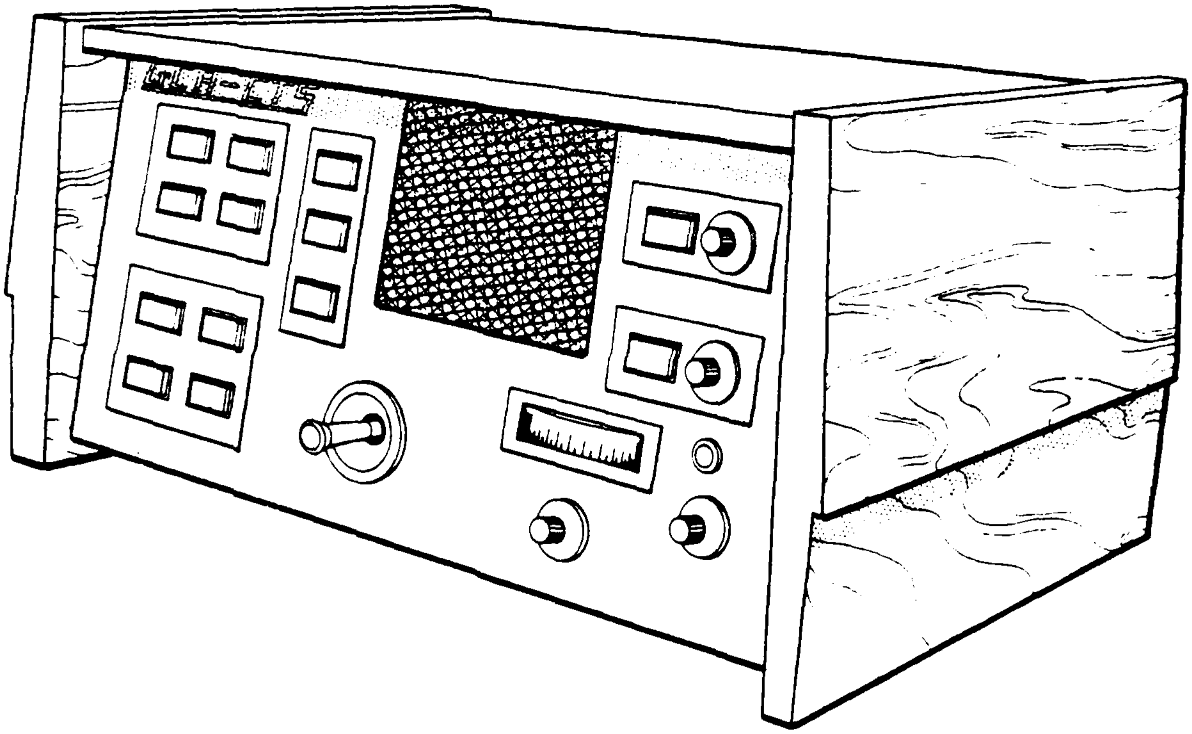


Figure 5. Trainee Panel, Front View

TABLE 4. TRAINEE PANEL INTERFACE ELEMENTS

Item	Description	Function
Servo Mechanism	A deflectable joystick extending 2" perpendicularly from panel Can be deflected 1-1/2" horizontally and/or vertically. Has spring return Has pushbutton in tip	Simulates antenna servo controls.
Radio Frequency Selection	A total of four square pushbuttons two for 270.8 megahertz, two labeled 318.8 megahertz. Buttons are alternate action. For each frequency one button is split with amber and green light table segments The other button has only an amber light.	The radio frequency panel consists of two sets of two button lights. In each set the first button light is the frequency select button. When the particular radio frequency is available for use the button light is not lit. When the frequency is in use the button is amber and the controller will hear an alarm in his/her headset if he/she selects the frequency. When the frequency is available and the controller selects it, the button light turns on and stays green until deselected. The second button light of the set is the monitor button light which the PAR controller selects when he/she wishes to monitor the communications between the pattern controller and the aircraft pilot. The amber light within the button comes on and stays on until the button is deselected.
Inter-Controller Communications System (ICS)	A total of four square alternate-action pushbuttons. Each contains an amber light which may flash or be steady on or off. Labels are: 3 5 7 SUPER	The ICS is used by the PAR controller to communicate with the pattern controller or to monitor approaches conducted by other positions. The button light must be depressed in order for the PAR controller to communicate with the pattern controller. The button light will be illuminated as an amber source when the button is depressed and will remain on until the button is deselected. 3 5 7 are for other controllers. SUPER is for the supervisor (instructor).

TABLE 4. TRAINEE PANEL INTERFACE ELEMENTS (CONT)

Item	Description	Function
Clearance System	One square alternate-action lighted pushbutton and two separate square lights. The switch light is white and is labeled "REQUEST." A green light is labeled "CLEARED." A flashing red light is labeled "W/O."	The tower clearance light system consists of two button lights and one individual light. The system is used by the PAR controller to request landing clearance from the tower and by the tower to inform the controller of landing clearance or to cancel that clearance. The button light is illuminated as a white source when the button is depressed. The cleared light is a green light that indicates that the tower has granted the aircraft clearance to land. The second light is a red flashing button light that indicates that the tower has cancelled the landing clearance. In addition to the flashing red light, an auditory alarm is also activated when the tower cancels clearance. Both go off when the button is pressed.
Volume Level Meter	A horizontal meter. Calibrated from 0 to 10. 0 - 4 Colored White 4 - 8 Colored Green 8 - 10 Colored Red	Shows trainee's voice level. Should be in green region when "FIVE" is spoken.
Volume Control	A knob with the numbers 1 through 5 on the skirt	Used to set the proper level on the Volume Level Meter
Audio Output Controls	Two alternate-action switches, lighted white when selected. One is labeled "HEADPHONE," the other is labeled "SPEAKER." Two knobs each having 300° of rotation. Each is labeled "LOUDNESS."	These allow either the headphones or the speaker to be energized at the level desired.
Speaker	A 4" dynamic loudspeaker	Audio output
Headset Jack	A 4-circuit jack for headset (microphone plus earphones).	Audio output

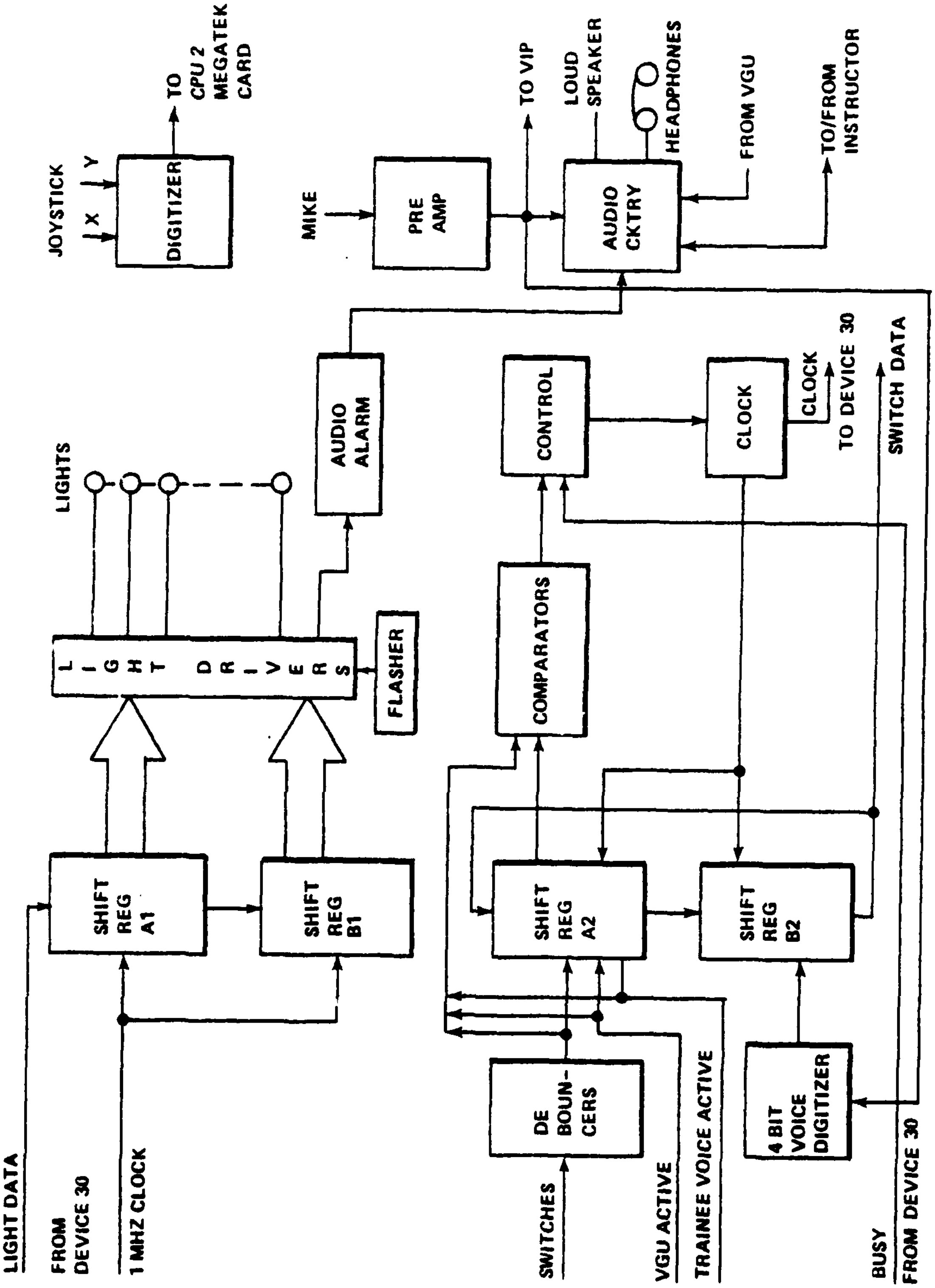


Figure 6. Trainee Panel Block Diagram

THE INSTRUCTOR PANEL

This unit is part of the instructor station. It was designed by Logicon. It is roughly similar to but much simpler than the trainee panel.

Figure 7 is a view of the front panel. Table 5 describes each of the elements on the front panel.

It is connected to the system controller by a single 100-foot cable.

All power for the instructor panel is provided by two external DC power supplies through a single rear-mounted connector.

THE JUNCTION PANEL

This panel, located in the rear of the system controller, has been included to consolidate signals into a minimum number of cables running to the trainee panel and the instructor panel. It consists of five cable receptacles and a set of test points, properly wired.

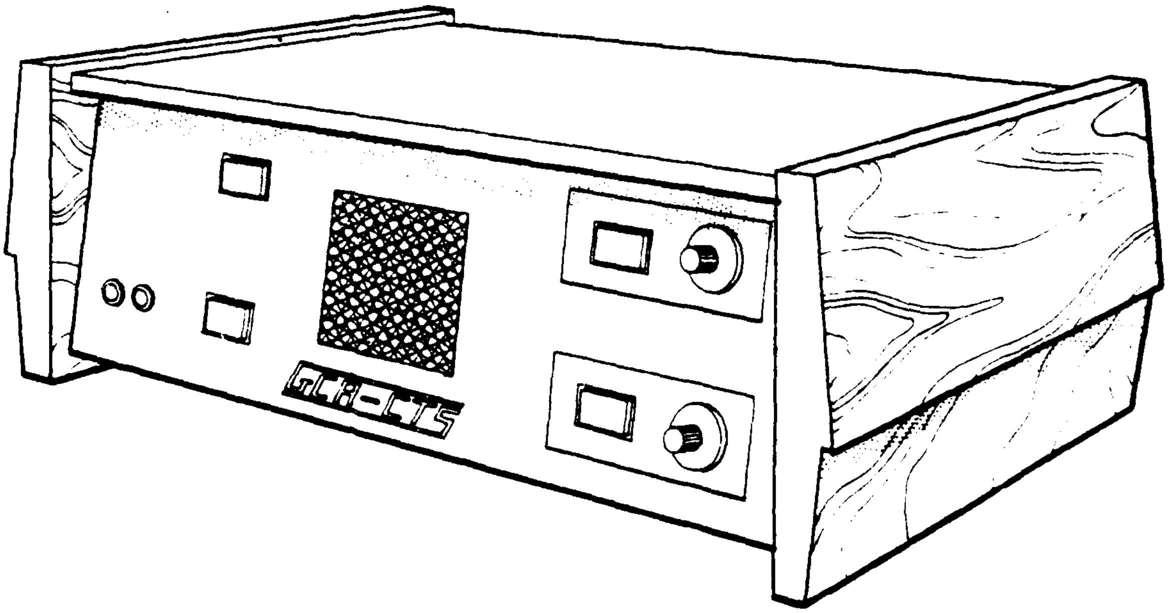


Figure 7. Instructor Panel, Front View

TABLE 5. INSTRUCTOR PANEL INTERFACE ELEMENTS

Item	Description	Function
Audio Output	Two alternate-action switches lighted white when selected. One is labeled "HEADPHONE" and the other is labeled "SPEAKER." Two knobs each having 300° of rotation. Each is labeled "LOUDNESS."	These allow either the headphone or the speaker to be energized at the level desired.
Headset Jack	A dual jack to accommodate micro-phone and headphones	Audio output
Speaker	A 4" dynamic loudspeaker	Audio output
ICS	A square momentary switch, lighted. The light may be off, flashing amber or steady amber.	This button must have been depressed to signal the computer to turn the light on steady. It must be on for the instructor to talk with the trainee. (See note below.)
Audio Monitor	A square momentary switch, lighted	Repeatedly pressing the switch will toggle the light on and off. When light is on and trainee does not have SUPER depressed, the instructor can monitor VGU, trainee or device 31 playback.

Note: Unit can be in ICS or Audio Monitor mode or neither, but not both.

SECTION IV

SOFTWARE ENVIRONMENT

The software environment includes both the system support software used for normal operations and the diagnostics used for preventive maintenance and troubleshooting.

SYSTEM SUPPORT

GCA-CTS relies on vendor-supplied support software. Specifically, it takes advantage of the many features of the Real-Time Disk Operating System (RDOS) and uses the Fortran language. Most of the coding is in Fortran 5, although some Fortran-compatible assembly language code is included. GCA-CTS makes use of a vendor-supplied graphics software package. The Data General RDOS, Fortran 5 and Macro Assembler are described briefly in the paragraphs that follow. A discussion of the Megatek graphics library routines is also included.

RDOS. RDOS was shown in the laboratory version to be capable of meeting the demanding real-time response requirements imposed by the GCA-CTS. It has the capability to schedule and allocate control to many different program tasks to provide simultaneous use of system resources and thereby maximize the efficiency of program operation.

The RDOS executive constitutes the main framework of the operating system, and it is resident in main memory at all times. Functions performed by this resident portion of RDOS include interrupt processing, overlay and buffer management, system call processing, and device interrupt servicing. Other modules of the system are brought into main memory from disk storage, as they are required to perform specific functions such as device initializations, file maintenance operations, and spooling control. In addition, the mapped RDOS used by GCA-CTS supports mapped memory addressing. The memory allocation and protection (MAP) unit provides a hardware separation of operating system areas from user address space. Moreover, it extends the maximum core configuration for a single CPU from 32K total to up to 32K for the resident operating system and up to 32K directly addressable by the foreground partition and 32K directly addressable by the background user. In a mapped system, two addressing modes exist. In the first mode, absolute mode, only the lower 32K is directly addressable and the mapping device is not used. RDOS resides in these low physical memory locations and executes in absolute mode.

The second mode is called mapped, or user mode. In user mode up to thirty-two 1024_{10} word blocks of memory are mapped by the management unit to produce an apparent (logical) 32K continuous address space. Any program operating in user mode uses a complete logical address space including its private page zero and extending through its upper memory bound. This upper bound is determined by the requirements of the individual program and it may extend as high as 32K. The operating system is responsible for assigning

free memory from its available pool to each user program prior to its execution. The technique used to manage the mapping unit and the construction of the user program in logical address space is also the responsibility of mapped RDOS.

Although mapped addressing extends the total amount of resident memory, it does not itself permit any single user program to exceed 32K words of memory. Since this restriction is unacceptable to some application programs, including GCA-CTS, RDOS provides two facilities for accessing the extended address space above 32K. Both virtual user overlays and window mapping create extended address space by storing data into memory blocks outside the 32K address space directly accessible by the user. When this program material is to be accessed, the desired blocks are remapped into the user's address space by enabling the memory management unit.

RDOS also provides the capability to bring in parts of a program from the disk as they are needed. The RDOS system can reserve portions of user address space for this function and divides it into fixed-length partitioned core storage areas which form a repository for programs of a limited size. This allows the RDOS user to segment a larger program into one or more parts which fit into the fixed-size core areas at execution time. These program segments are called user overlays and are stored on disk in core image format to facilitate rapid loading when their execution is required.

Other features of RDOS include full I/O support for a wide range of peripherals including the disk, CRTs, the printer, and the IPB.

An important function of any real-time operating system is the efficient handling of input-output operations. Optimum usage of matching devices and central processor time in the accomplishment of tasks is a major reason for designing and implementing a multitasking system. Since I/O devices are slow compared to the internal speed of the computer, they must be programmed to overlap their operations with computations, when possible, in order to increase usable CPU time by allowing one task to operate while I/O is in progress, to greatly increase efficiency of I/O operations, and to provide more throughput of data by removing bottlenecks caused by slow peripherals. The responsibility of RDOS I/O control is to react during normal program execution to the structuring of I/O requests, making assignments of requests to machine devices when they are idle, and queuing requests for devices which are busy. Through the queuing facility, RDOS makes it possible to achieve maximum and continuous overlap of many tasks without direct intervention by the tasks themselves.

The concept of a task is central to an understanding of GCA-CTS operation both at the level of I/O handling and at the applications program level. A task is a logically complete, asynchronous execution path through a program, subprogram, or overlay which demands use of system resources (usually CPU control). Many tasks may be directed to operate in a single re-entrant path, and each of these tasks may be assigned a unique priority. One real-time program may have from several to a virtually unlimited number of logically distinct tasks. Each task performs a specified function asynchronously and in real-time. CPU control is allocated by the RDOS task scheduler to the highest

priority task that is ready to perform or continue performing its function. This system scheduler and its associated routine together support the high level Fortran 5 tasking calls through the Universal Multitasking Interface (UMTI).

In addition to these runtime support functions, RDOS provides a powerful Command Line Interpreter program and also editors, compilers, assemblers, and debuggers which allow interactive software development to proceed in an efficient, user-oriented way.

FORTRAN 5. Fortran 5 is an ANSI standard superset of Fortran developed by Data General. In addition to standard Fortran, it includes the following features:

Full mixed mode numeric conversion,

Acceptance of any expression as a control variable or parameter in a DO statement or DO-implied list,

Generic library functions,

Declarations that may appear anywhere in the program,

No reserved words or reserved function names,

All blanks ignored, except in Hollerith constants,

IMPLICIT statement for data typing.

Compiler. The Fortran 5 compiler provides a mechanism for generating very efficient object programs from programs written in a superset of the Fortran language.

The efficiency of Eclipse Fortran 5 code derives from its full use of the powerful Eclipse instruction set and from the optimization of the generated code. Subscript computations, type conversion, comparisons, many library functions, etc., are generated in-line and can thus take full advantage of the compiler's interstatement optimization and ability to search for common expressions that need to be evaluated only once. Local optimization includes the following:

Multiplication of an integer by a power of 2 is performed by shifting.

Redundant operations, such as addition of the constant 0 or exponentiation to the constant power of 0 or 1, are eliminated.

The compiler takes advantage of the associative or distributive properties of operators by reordering or eliminating some operations.

Exponentiation of variables by positive integer constants is performed by in-line multiplication.

Variable and expression values may be assigned to and remain in registers throughout a portion of the program.

Floating point operations are optimized for effective use of floating point hardware. Hardware floating point operations will be performed asynchronously when possible.

Runtime Support. The Fortran 5 runtime library package supplies routines for performing integer, single and double precision real and complex mathematical operations, routines that perform formatted and unformatted input and output, and file creation and maintenance functions, routines that provide interfaces to system facilities, and routines that create and maintain a multitask environment and provide overlay management facilities. All routines in the library are re-entrant, permitting one or more tasks to enter and execute a routine before prior executions are complete. This means that many tasks can share a single copy of a routine, and significant core savings result. GCA-CTS will also make use of the Load-On-Call Overlay Facility (LOCO) which automatically loads and releases overlays as required.

Runtime Environment. Figure 8 illustrates the runtime configuration of main memory for a multitask Fortran 5 program. RDOS resides in a separate address space and so is not shown. Three levels of data must be distinguished in this environment:

Per-ground data are common to the entire ground; that is, data which are global to all tasks (if more than one task exists) but which are local to a given ground (if foreground and background both exist in a mapped system). They include all linkage to runtime routines in page zero, certain information maintained by RDOS about the ground, common blocks and static storage, the runtime file table, and all executable code.

Per-task data are particular to a given task. They include task status and priority; the values of the accumulators, carry, and program counter; the state of the floating point unit; the Fortran 5 state variables; I/O control information; and the task's runtime stack.

Per-routine data are particular to a single execution of a runtime routine. They are stored in a distinct portion of a task's runtime stack known as a frame. Using different stack frames for different activations of the same routine makes the routines re-entrant, as required for multitask operation.

Figure 8 shows how these data are stored. Shown are:

a. Page zero, containing the task state variables (.FP,.SP,.SSE,.RP, and .GP); and locations reserved by the hardware and by RDOS.

b. RDOS tables, including the User Status Table (UST) and an (optional) overlay directory, which contain per-ground data; and the pool of task control blocks (TCB), each of which may contain per-task data for a single task.

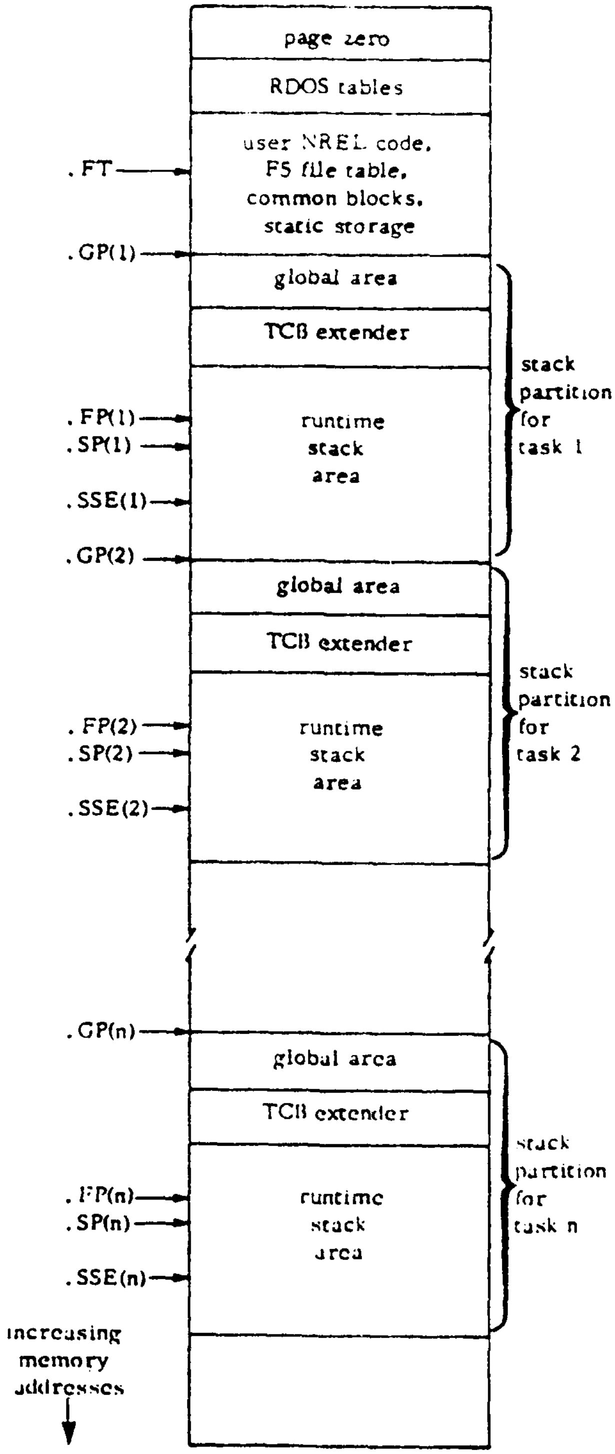


Figure 8. Runtime Memory Allocation

c. User normal relocatable code (including the Fortran 5 main program, subprograms, runtime library routines, and routines from the system library), the Fortran 5 runtime file table (pointed to by page zero location .FT), common blocks, and static storage. All data in this area are per-ground data.

d. Per-task global area, containing mainly I/O task control information. The page zero state variable .GP points to the start of the global area for the currently executing task.

e. For multitask environments, a task control block extender, containing additional per-task information which must be maintained for a Fortran 5 task, such as the state of the floating point unit and the Fortran 5 state variables. The word at offset TELN of a task's TCB points to its TCB extender.

f. The runtime stack area for a given task, used by compiled Fortran 5 programs and runtime routines for local data storage.

As shown in the figure, each task in a multitask environment has its own per-task global area, TCB extender (where values for its state variables are stored when it is not executing; just as the values of its accumulators, carry and program counter are stored in its TCB), and runtime stack area. By default, the memory available at initialization is divided equally into as many stacks as there are TCBs. Fortunately, a partition macro is available which can be used to specify the number and the size of stack partitions. This is of critical importance in an environment like the GCA-CTS where many tasks compete for limited core resources.

MACRO ASSEMBLER. In general, an assembler allows source programs to be written using familiar characters to create symbols that are meaningful to the programmer. The assembler processes these source programs to produce object programs in machine language, meaningful to the computer. To do this, the assembler simply substitutes a numeric code for each symbolic instruction code and a numeric address for each symbolic address. The Data General macro assembler includes the following added features:

a. Expanded expression evaluation that provides for explicit as well as implicit precedence. The class of operators includes relational operators.

b. A powerful macro facility which allows complete recursion as well as nested macro calls.

c. An assembly repeat feature for producing many lines of source from a simple repeat construct. This facility also encompasses conditional assembly. Conditionals may be nested to any depth.

d. An assembly suppression feature that allows the programmer to suppress assembly until a given label is encountered.

e. The assembler can generate a three-digit number to replace a symbol anywhere in assembly code. Thus, the digits may be part of a symbol or number or may stand alone. The feature is useful, for example, in providing unique labels during table generation.

f. A class of special symbols having a value (like ".") related to an internal assembler variable. This class of symbols allows the user to determine useful information such as the number of arguments specified by a current macro call. Further, many pseudo-ops have a value associated with them and, using the proper syntax, may be used within expressions.

g. Literal references by any memory reference instruction. All literals will be optimally resolved in page zero. Literals are not restricted to absolute numeric quantities and, in fact, may consist of any legitimate expression.

h. The assembler has the facility to generate three-character alphanumerics for each occurrence of the character $ within a label. The facility is implemented in such a way that, for example, unique labels can be generated within nested macros.

MEGATEK GRAPHICS LIBRARY. The Megatek display processor series 5000 provides the support for the graphics portion of the GCA-CTS. The Megatek consists of a processor and a large cathode ray tube (CRT). It contains no user memory, that being provided by the Eclipse and accessed through a DMA cycle-stealing device. Software provided with this system was designed for an unmapped machine and the routines were not compatible with Fortran 5. Therefore, it was necessary to modify the software to perform in a mapped Fortran 5 environment.

The Megatek features vector graphics, full translation in the X- and Y-planes, and a hardware generated character set. Pictures are created within a display list with a series of microprocessor instruction codes. These codes inform the Megatek processor of the desired location of the CRT beam, and whether the vector to be drawn is visible or blanked. It is possible to append, insert, delete, or write over any picture component within the list. The screen can be referenced in screen units (4096 x 4096) or units defined by the user. Pictures can be modified dynamically, changing and replacing pictures very rapidly.

For optimal use of the Megatek processor, a Megatek graphics package is being used. This package contains a series of Fortran-callable subroutines which build the display list, thus making Megatek usage extremely simple. There are routines to draw lines, move the CRT beam without drawing lines, translate pictures, including rotation, and enlargement or shrinkage of pictures. There are also routines to draw the hardware generated characters, and activate the joystick. All of these access the display list, either directly, or through a lower level assembly language routine. Other programs provide support for the main routine. These include programs to convert floating point numbers to ASCII characters, to change the limits of a picture, or of the joystick, and to provide the coordinates of the joystick position.

DIAGNOSTICS

A wide range of diagnostics was supplied with the GCA-CTS. These can be used for routine preventive maintenance, to ensure system integrity after shipping and installation, and for troubleshooting hardware failures. Diagnostics provided by vendors and by Logicon are described in the following pages.

DATA GENERAL CORPORATION'S DIAGNOSTIC OPERATING SYSTEM (DDOS). DDOS is an operating system that has been developed by Data General Corporation to provide an efficient and systematic method of running diagnostic tests on DGC processors and peripheral equipment.

DDOS may be used most efficiently for problem isolation and detection by following simple procedures.

A special debugger program has been included with DDOS, which allows the operator to isolate sections of a diagnostic by the setting up of breakpoints.

DDOS is available in either cartridge disk, diskette, or magnetic tape form.

The minimum equipment requirements for using DDOS are a DGC processor, a terminal, and a magnetic tape, diskette or disk drive. DDOS contained on magnetic tape will run successfully in processors having a minimum of 4K words of memory. DDOS contained on diskette will run in processors having a minimum of 8K words of memory. Some diagnostics require more space than the minimums specified above. If a diagnostic should be scheduled that is too large for the memory in the processor, an error message will be printed on the terminal to so inform the operator. For reference purposes, it is necessary to have a listing of each diagnostic program that will be run.

The operator controls device testing with simple on-line commands issued to the processor through a terminal keyboard. These are interpreted by the DDOS monitor to allow the operator to load and run any diagnostic contained on the tape or disk. In addition to specifying diagnostics, commands are used to determine which I/O devices will be tested. Certain commands also allow the operator to run a data channel test concurrently with a diagnostic.

DDOS constructs a table in memory, called the equipment table, containing the mnemonic and device code for each piece of equipment on the processor data bus. Devices contained in this table will be tested automatically when certain commands are issued. Devices which are not included in the equipment table automatically may be added to it with simple DDOS commands. DDOS will only support DGC diagnostic tests. Any user written diagnostics will have to run separately from this operating system.

Table 6 gives a list of the Data General diagnostics which are applicable to the GCA-CTS hardware.

TABLE 6. DATA GENERAL DIAGNOSTICS

Title	Function
ECLIPSEA: Central processor diagnostic part 1	Tests arithmetic and logical operations.
ECLIPSEB: Central processor diagnostic part 2	Tests bit manipulation instructions, accumulator compare and logical shift instructions.
ECLIPSEC: Central processor diagnostic part 3	Tests logic of memory reference instructions, auto-increment and decrement, etc.
ECLIPSED: Central processor diagnostic part 4	Tests stack manipulations, extended operations, etc.
ECLIPSEE: Central processor diagnostic part 5	Tests logic of two-word instructions, etc.
ESPCLEX: Special exerciser	Test all instructions, mapped and unmapped, with ERCC option.
ECLIPSE**: Exerciser parts 1-9	Tests reliability of CPU instructions.
EIMRT S: Multi-program reliability - short	Tests CPU, memory, floating point, map and character instructions.
EIMRT L: Multi-program reliability - long	Like EIMRT S, but also tests primary disk and printer.
EIMRT P: Multi-program reliability - peripherals	Like EIMRT S but also exercises peripherals.
EMMPUA, EMMPUB: Memory allocation and protection unit test	Tests the MAP feature.
ECLSC: Semiconductor memory test	Tests semiconductor memory.
EXMEM: Extended memory exerciser	Tests memory, taking MAP and interleaved memory into account.
EMLER: ERCC diagnostic multilayer CPU2	Error checking and correction test.
EPFAIL: Power shutdown test	Tests power monitor and auto restart option.
IPBR: Inter-processor bus reliability.	Tests the various types of IPB transfers.

TABLE 6. DATA GENERAL DIAGNOSTICS (CONT)

Title	Function
IPBD: Inter-processor bus diagnostic	Tests a single IPB board.
EIOA, EIOB: I/O tests	Verifies operation of the I/O features, I/O bus, interrupt and data channel and VCT instruction.
40DI: 4010/4023 or 4077/4078 diagnostic	Tests specific I/O boards.
ETTY: Teletype test	Detects malfunctions in the teletype logic.
LCD: Video display test	Checks 6052, 6053 video displays.
RTCTST: Real-time clock test	Real-time clock maintenance.
CDF: Cartridge/diskette formatter	Formats disks and diskettes.
CDR: Cartridge/diskette reliability	Exercises disk controller and drives.
EIFUPX: Floating point firmware exerciser	Tests floating point instruction set reliability.
EIFPUD: Floating point firmware diagnostic	Detects failures in floating point unit.

Most of the DGC diagnostics consist of a series of simple tests, each of which sends a particular combination of input signals to a small portion of the unit under test, and performs some simple test on the output. Generally, each of these tests is initialized by a small subroutine. This subroutine sets the internal pass counters, that is, the number of passes to be made through the test, establishes the proper address for the diagnostic to jump to after each pass, and determines any other parameters necessary to run the test. Another small subroutine keeps track of the number of times the diagnostic has been run successfully. This subroutine is responsible for having the diagnostic jump to a particular address after it has been run the number of times that had been established with the internal pass counter. If the diagnostic has not completed the established number of runs, this small subroutine forces the diagnostic to begin again.

Simple Diagnostics. On some of the simple diagnostics, a failing test will simply cause the processor to halt. The accumulators will contain some information about the failure and the address lights will give the location of the failure. The user must consult the listing of the diagnostic test that was running to find the exact reason for the failure. A failure in this type of diagnostic almost precludes passing any other diagnostic.

Complex Diagnostics. On the more complex diagnostics the stop or halt instruction will be imbedded in a subroutine. This subroutine will force the diagnostic to print the memory location where the failure occurred and return control to the monitor program. If a command has been issued which causes all applicable diagnostics to run in sequence continually, the monitor will print the name of the diagnostic in which the failure occurred in addition to printing the location of the failure, each run through the entire list, excluding the first.

Overnight Testing. DDOS was designed such that control is returned to the monitor if an error occurs in a complex diagnostic so that the system will not hang up on a single failure when a relatively long run is attempted, such as overnight. When control returns to the monitor, the next scheduled test, if any, will be loaded and run. Consequently, to determine whether any errors have occurred, the terminal or line printer output must be examined.

Program Modes. Diagnostics are executed under DDOS in one of four modes: auto, semi-auto, manual, or debug. The primary differences between the modes lie in the associated operator communication for each diagnostic.

Auto Mode. In auto mode DDOS compares its equipment table against the equipment requirements of the test programs as shown in the directory, and sequentially executes those programs that exercise the devices on the machine under test. Each test program is loaded and executed automatically. At its conclusion the test program returns control to DDOS so that the next program can be run. No operator communication is required after the initial command.

Semi-Auto Mode. If communication with the operator is required, (to establish what surfaces of multi-surface disk are to be tested by a disk reliability program, for example) the program may be run in semi-auto mode. Programs to be run in semi-auto mode will not be run in auto mode; however, in all other respects operation is similar. One or more test programs may be specified in the initial command to DDOS and these programs will be executed sequentially with return being made to DDOS after each program ends.

Manual Mode. Manual mode is used when a return to DDOS is not desired. The initial command results in DDOS loading the program and starting it; however, at the conclusion of the test, the test program loops back to the start of the test. This mode is useful when the operator is troubleshooting a machine failing a particular test. The LOAD and CLOAD commands cause DDOS to operate in manual mode.

Debugger Mode. If the DEBUG command has been given, the debugger is loaded at location $30{,}000_8$ for the Eclipse line processors, then the test program is loaded. DDOS transfers control to the debugger rather than to the starting address of the test program. This procedure is useful for things other than debugging. For example, a program to test the disk drive can be loaded with DEBUG and the program will not run until the operator starts it. This gives him time to dismount the DDOS disk and mount a scratch disk.

Once the test program is started from the debugger, operation is identical to manual mode. The test program will loop on completion without returning to DDOS. It should be noted that the test program is free to write over the debugger, and if this occurs, the debugger becomes useless after the program starts. Individual diagnostic listings should be consulted to determine whether or not the debugger will be overwitten.

MEGATEK DIAGNOSTIC PROGRAM. A vendor-supplied diagnostic program exercises the graphics display processor. It provides a variety of test patterns which can be used to observe picture alignment, refresh rate changes, blinking, and the various levels of intensity. Hardware faults can be traced by observing their effect on the test patterns. These test patterns are also used in display alignment procedures. The diagnostic is supplied as a stand-alone program which is BOOTed in from the disk or diskette.

TALLY DIAGNOSTIC. The Tally printer has a self-test mode of operation which is selected by a switch setting on the side of the unit. The test exercises every character in every position. Detection of functional failures requires only a cursory glance at the one page test pattern.

DAILY OPERATIONAL READINESS TEST (DORT)

DORT was designed to test each GCA-CTS device before the GCA-CTS program is executed. DORT gives the user a demonstration that each device is working properly by producing visible output from the device which can be evaluated by the user or by exercising the devices and displaying the results.

PROGRAM OVERVIEW. DORT consists of two programs, DORT1.SV and DORT2.SV which operate simultaneously in CPU1 and CPU2, respectively. Interprogram communication is accomplished via the IPB. DORT1 can test the Votrax, the instructor panel, and initiate a test of IPB. DORT2 can test the Votrax, the student panel, the voice digitizer, the Megatek display processor, the TTI 500 voice input processor, the High Speed Correlator, and the IPB. Figures 9 through 14 are block diagrams of these programs. In the block diagrams, double lines are used to show modules for which a more complete block diagram is given. Dashed connecting lines are used for tasks; solid connecting lines are used for subroutines and functions.

The Votrax Test. The Votrax test is available from both the student and instructor stations to ensure that the Votrax itself and all of the speakers and headphones are working properly. During the test, DORT will instruct the Votrax to speak twice, once while the user is listening to the speaker in the panel, and once while the user is listening through the headset. The user is then asked to evaluate the Votrax's performance.

The Panel Test. The panel test is available from both the student and instructor station to permit the testing of all panel buttons. DORT activates the panel and manipulates the panel lights so that the user can evaluate panel performance.

The Digitizer Test. DORT tests the Digitizer by prompting the user to speak a phrase which is recorded and played back for the user to evaluate.

The Megatek/Servo Test. For this test, DORT draws a GCA-CTS type picture and activates the joystick monitor so that the user can evaluate the performance of these devices.

VIP/HSC Test. The VIP test evaluates the performance by counting the number of times each individual feature is set when the Votrax speaks to the VIP and comparing the counts to "normal" counts that were collected earlier. The High Speed Correlator (HSC) is tested by comparing the result of correlating random numbers computered by a software simulation of the HSC with the result returned by the HSC itself.

IPB Test. DORT tests the IPB by attempting to establish an active line of communication between CPU 1 and CPU 2. If contact is established, the IPB will be used by CPU 2 to request CPU 1 to perform certain functions that produce visible results at the student station but can only be performed by CPU 1.

DORT ERROR ANALYSIS. DORT produces two error files, DORT$ERRORS and DORT2$ERRORS, which are created by DORT1.SV and DORT2.SV, respectively. Figures 15 and 16 are examples of these files. If an error is discovered, DORT will issue a warning at program termination. The user must then track down and correct any errors.

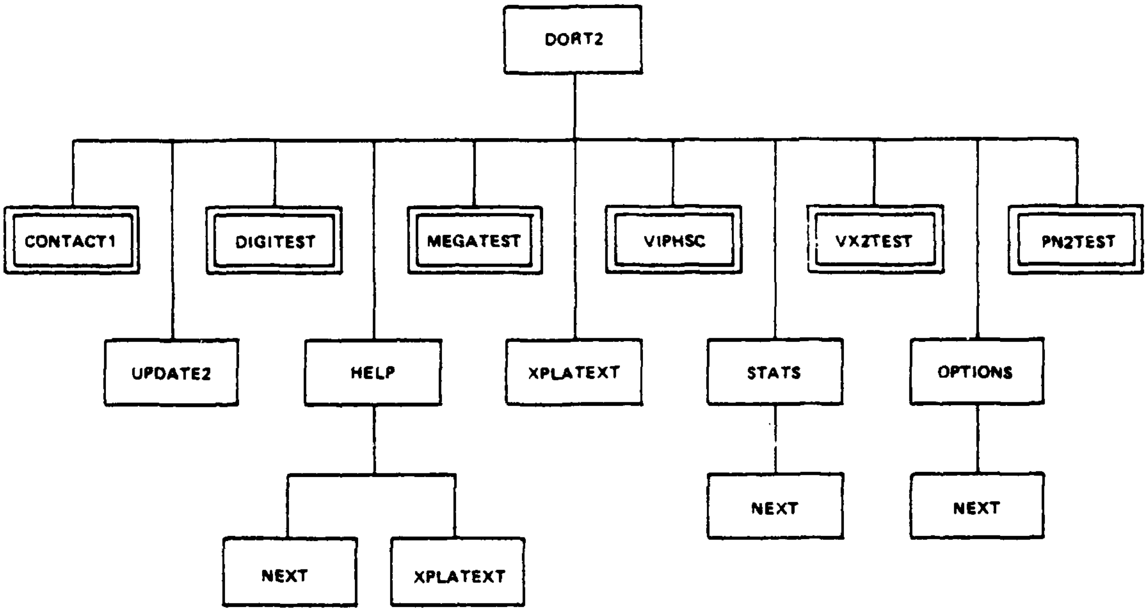


Figure 9. Overview Block Diagram for Main DORT Program on CPU 2

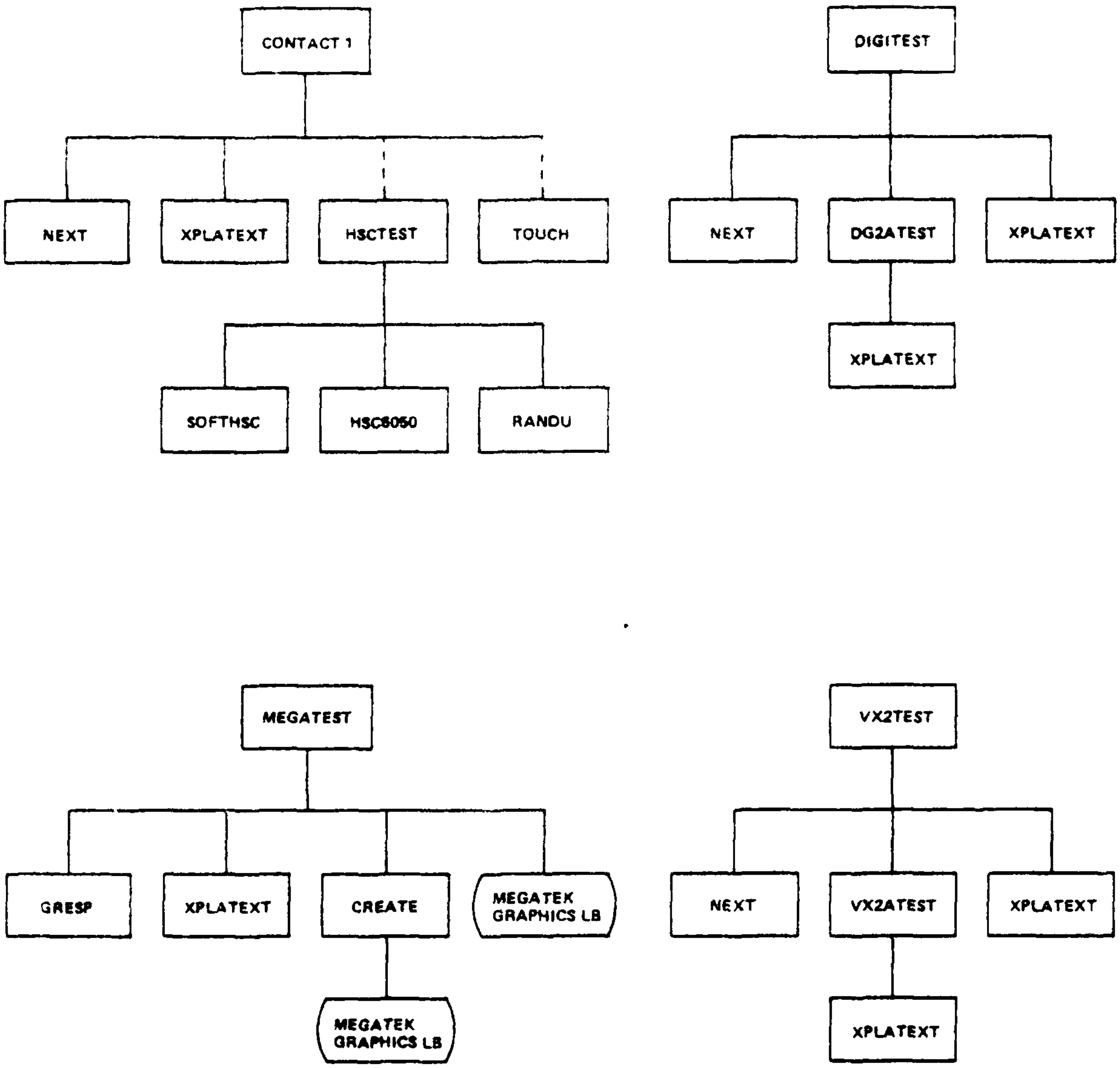


Figure 10. Specific Block Diagrams for Test Routines of DORT CPU 2 Including Digitizer, IPB, Megatek and Votrax

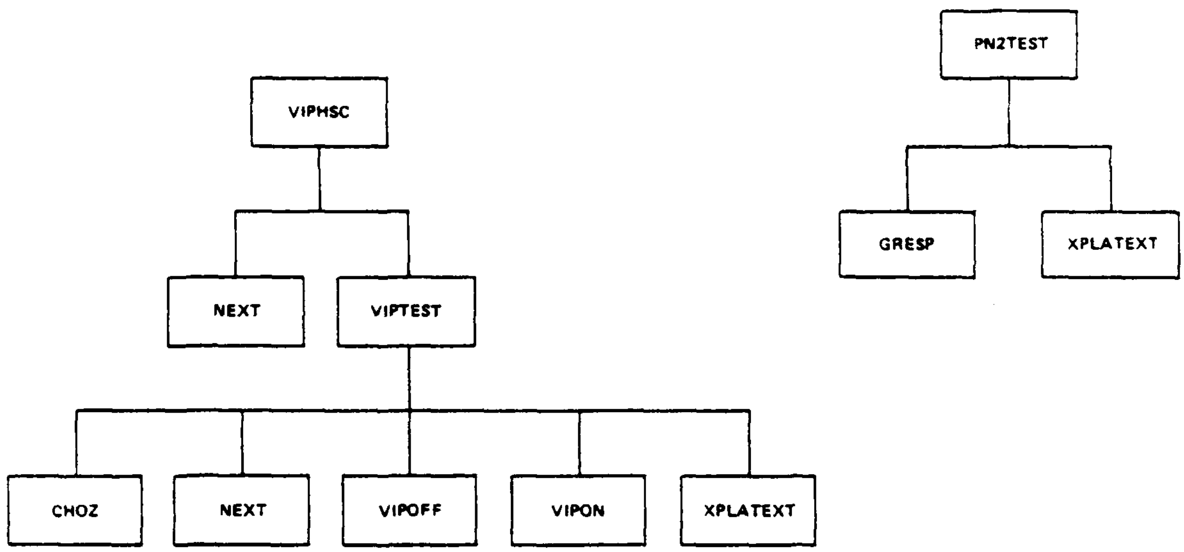


Figure 11. Specific Block Diagrams of DORT CPU 2 for Trainee Panel and Speech Recognition

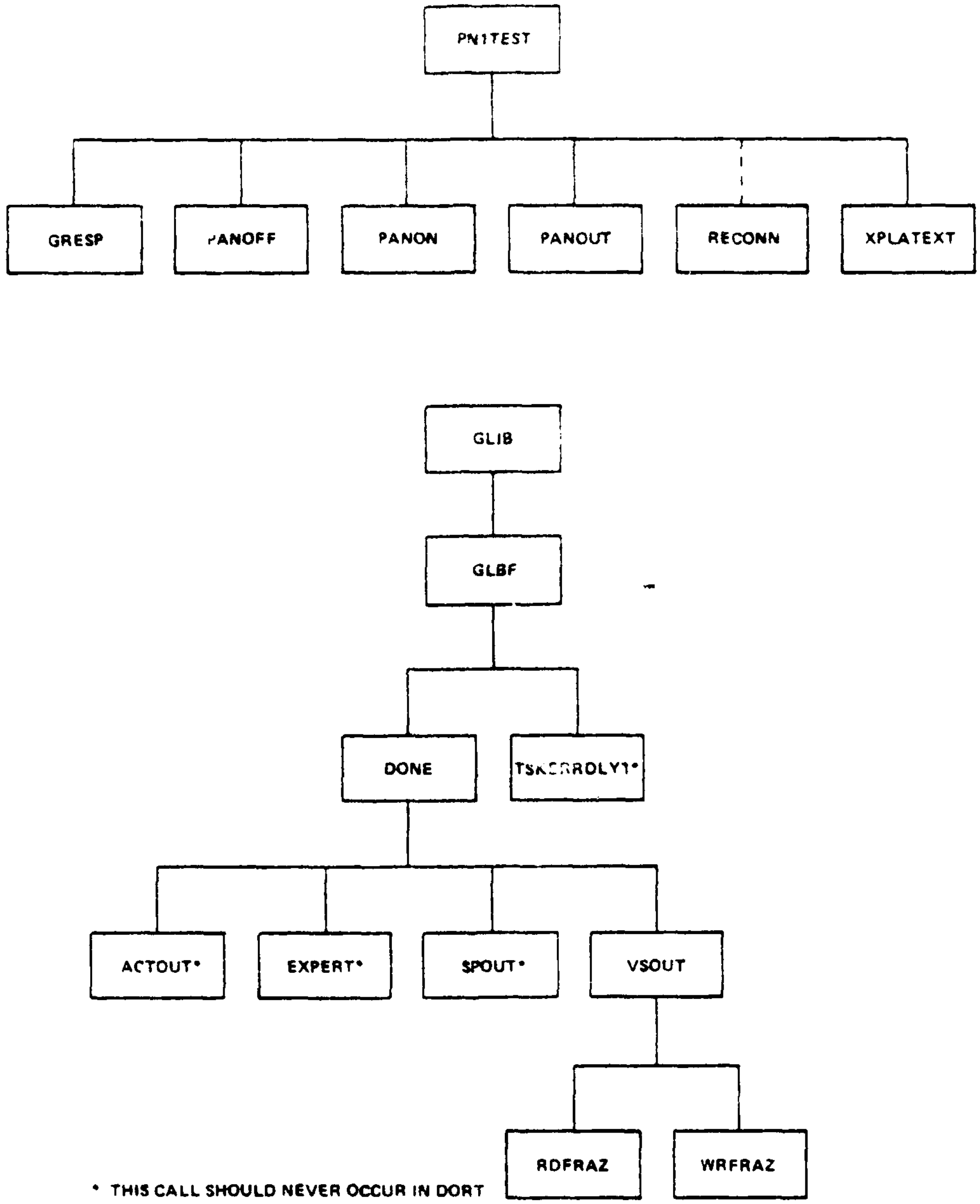


Figure 12. CPU 1 DORT Block Diagrams for Instructor Panel and Speech Generation

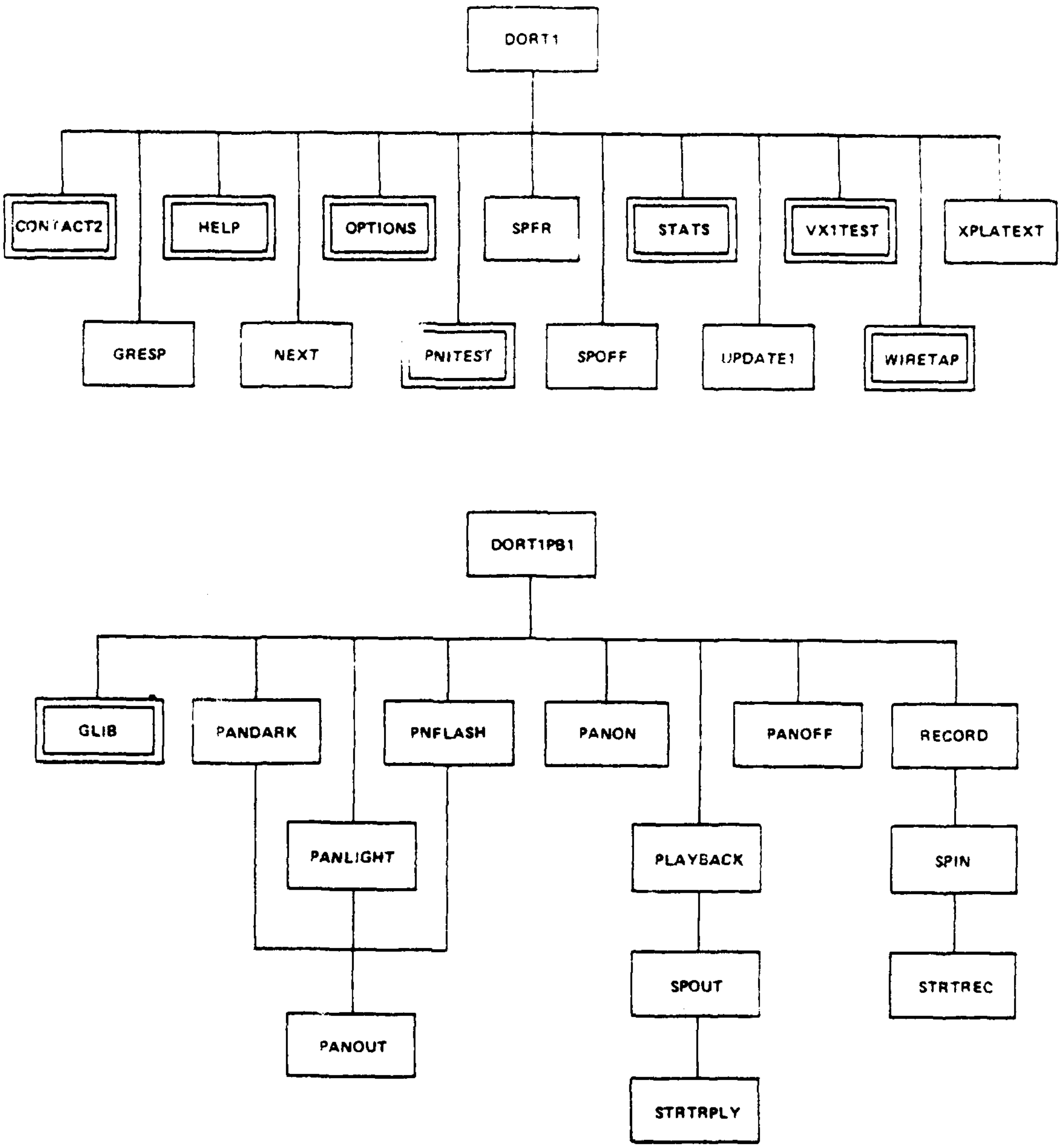


Figure 13. Overview of DORT On CPU 1 with IPB

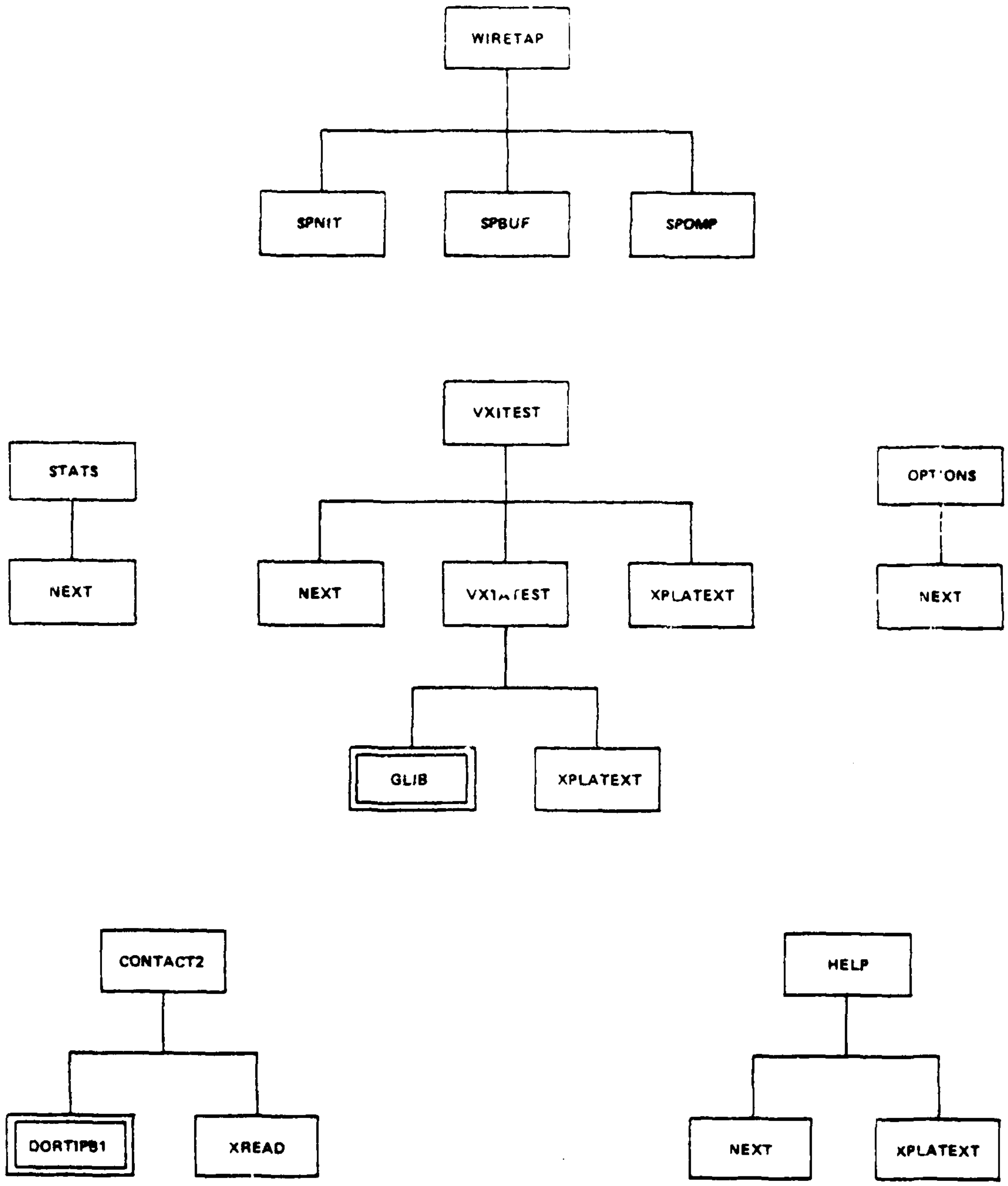


Figure 14. Block Diagram for DORT CPU 1 for Digitizer, Votrax and Utility Routines

	VALID**	WORKING*	ATTEMPTS	FAILURES
VOTRAX (computer generated voice)	1	0	2	2
PANEL (student/instructor panels)	1	0	1	1
IPB (establish contact with CPU 2)	1	1	1	0
DIGITIZED VOICE (recording/playback)	0	1	1	0

**A 1 in the valid column means the test should appear in the list of available options, a 0 means that you must complete other tests before attempting this one.

*A 1 means that the device has been tested and functioned properly, a 0 means that the device is not working properly or has not been tested.

Figure 15. Example of DORT Test Results Summary for Side 1 Tests

	VALID**	WORKING*	ATTEMPTS	FAILURES
VOTRAX (computer generated voice)	1	1	1	0
VOTRAX (results of side 1 tests)	1	1	1	0
PANEL (student/instructor panels)	1	0	0	0
PANEL (results of side 1 tests)	1	1	1	0
DIGITIZED VOICE (recording/playback)	1	0	0	0
MEGATEK/SERVO (test radar display)	1	0	0	0
VIP/HSC (voice recognition)	1	0	0	0

**A 1 in the valid column means the test should appear in the list of available options, a 0 means that you must complete other tests before attempting this one.

*A 1 means that the device has been tested and functioned properly, a 0 means that the device is not working properly or has not been tested.

Figure 16. Example of DORT Test Results Summary for Side 2 Tests

SECTION V

APPLICATIONS SOFTWARE

OVERVIEW

The Training/Functional Design Report described a system which would both provide instructional materials and an environment for practicing GCA control skills. From the trainee's perspective, the GCA-CTS has five major modes of operation:

a. Phase 1

b. Phase 2

c. Phase 3 and P-run

d. Replay

e. Demonstration

Reviewing briefly, phase 1 provides multimedia presentations and demonstrations to teach the various topics in the syllabus and to elicit speech samples for vocabulary reference pattern creation.

Phase 2 is an optional freeze and feedback mode in which the student practices and the system freezes if a mistake is made on the new material. The mistake is explained and the student is given the opportunity to try again.

Phase 3 provides a simulated environment in which the trainee can practice the newly acquired skills and integrate them with old skills. This simulated environment is at first simplified somewhat, but as the student progresses through the syllabus it becomes more realistic. The P-run, or performance test, is the student's final examination. It is just like other phase 3 problems with the exception that special scoring options are available.

Several replay options are available for every phase 3 problem, thereby enabling the student to review his performance.

Finally, in the demonstration mode, the system conducts approaches utilizing a simulated final controller. This mode is used for instructional purposes in phase 1, and it also operates whenever the system is otherwise idle. This provides a natural transition to alignment checking procedures when the student signs on.

These modes of operation are woven together in the GCA-CTS as follows. The initialization routines start keyboard processing and IPB I/O routines, and they initiate the demonstration mode. When a student signs on to the system, the training control program takes over and selects one of the

instructional phases based upon the course syllabus and the student's progress to date. Each time the student completes a phase of instruction, the training executive regains control and selects the next mode. When the student signs off the system, the demonstration mode is initiated again.

The keyboard task remains active at all times, which allows the various special requests to affect the course of training.

The IPB I/O task is likewise always active, which reveals another layer of complexity within this simple scheme. The actual processing burden is divided between two computers which communicate via the full duplex lines of the IPB. These two processors, dubbed CPU 1 and CPU 2 in previous sections, can be thought of as the training controller and the speech recognition and display processor respectively. CPU 1 is the master and is responsible for controlling the modes of operation described above. CPU 2 is devoted primarily to processing speech input data from the Threshold 500 for voice data collection and speech recognition and to display processing. It also accepts keyboard entries from the trainee console and maintains IPB communications with CPU 1. Task allocation by CPU is transparent to the user with the exception that the startup protocol requires that the master computer be started first. From then on the GCA-CTS is one system, providing a range of training capabilities from computer-aided instruction through a final examination in a realistic radar environment.

The design goal of a flexible instructional system was deemed best met by the development of table-driven executives. A single executive is responsible for each phase of instruction and uses ASCII text files to provide the required variety of experiences. Changes to the course of instruction are implemented by simply editing the text files and do not require recompilation or reloading of the GCA-CTS executable routines.

The master file is the syllabus file which is used by the training control program. This syllabus contains an ordered list of file names and an indicator of the phase of instruction to which each corresponds. There is one of these [indicator, file name] entries for each phase of every task in the syllabus. Training control initiates the specified phase executive which then uses the information in the file to provide training or practice situations. These files are described in Appendix C.

Looking more closely at the requirements of the various modes of operation, it becomes apparent that they share many of the same functional elements. Table 7 shows these functional elements and the modes of operation to which they apply. The GCA-CTS routines which satisfy these requirements have therefore been designed to be general enough to operate as required in the various modes.

The discussion in the remainder of this report is organized according to a functional hierarchy which regards training control with its phase executives as primary, the major functions as secondary, and so on.

TABLE 7. FUNCTIONAL ELEMENTS OF THE GCA-CTS MODES OF OPERATION

Applicable Functions	Demonstration	Mode Phase 1	Mode Phase 2	Mode Phase 3, P-run	Replay
Voice data collection		X			
Speech recognition			X	X	
Speech understanding			X	X	
Aircraft, pilot, environment	X	X	X	X	
Radar	X	X	X	X	
Display	X	X	X	X	X
Controller models	X	X	X	X	
Performance measurement			X	X	
Keyboard input processing	X	X	X	X	X
IPB I/O processing	X	X	X	X	X
Trainee panel input processing		X	X	X	X
Trainee panel output processing	X	X	X	X	X
Votrax output processing	X	X	X	X	X
Speech digitizer input processing		X		X	
Speech digitizer output processing	X	X	X	X	X
User clocks	X	X	X	X	X

Communication between routines takes place by means of variables in labeled common and through disk files. These data structures are defined in Appendixes E and F, respectively. The common variable and parameter definitions given therein include the intertask communication message keys and event numbers.

The design of the GCA-CTS software is detailed in the subsections which follow. For each topic a narrative overview is provided with block diagrams where appropriate. Brief program descriptions for each routine are given in Appendix A.

INITIALIZATION

There are system initialization routines on both sides of GCA-CTS as well as individual initialization routines for the simulations and performance measurement. These individual routines are described within the sections that concern them directly. Initialization is achieved through the starting routines, two initialization routines and block data. On both sides all channels are opened, the IPB is initialized and console interrupts are disabled. Because of the possibility that extraneous data remains on the IPB, a test pattern is sent across the IPB by CPU 2 to be read by the initialization routines on CPU 1 for synchronization. CPU 1 also cues CPU 2 to start the Megatek display processor and to initialize pictures. CPU 2 starts the foreground listening task CKCMN. Block data is used to initialize display, recognition, keyboard and IPB variables.

TRAINING CONTROL

Training for the GCA-CTS system is divided into three phases in addition to a demonstration mode and a replay mode. The three phases are (1) instruction, (2) freeze and feedback and (3) practice and performance runs. During phase 1, the trainee is taught specific new behaviors based on the syllabus file residing on disk. Complete descriptions of the syllabus file and all other files can be found in Appendix C. All tasks must be covered in phase 1, since each task contains new information for the trainee. However, the number of problems per task can be manipulated to a degree dependent on the quickness of the trainee to master the given skill. Phase 2 provides feedback for the trainee by a special form of practice run. This run freezes when the trainee responds incorrectly to an event and explains the error via the CRT. The trainee is given the option of operating in the phase 2 mode before being tested. Phase 3 runs can provide realistic practice for the trainee, or they can be used to test the trainee for mastery of the given skill. In this phase, a realistic approach is executed by the trainee. Block diagrams are shown in Figures 17 through 27.

Aside from the restriction that the trainee must complete and master every task in the training program, the training process is extremely flexible. The trainee may choose not to practice in the freeze and feedback mode (phase 2). If a trainee performs poorly on a previously mastered task, a remediation task is selected via software to retrain the skill. In addition, the instructor can select any previously learned task or the next sequential

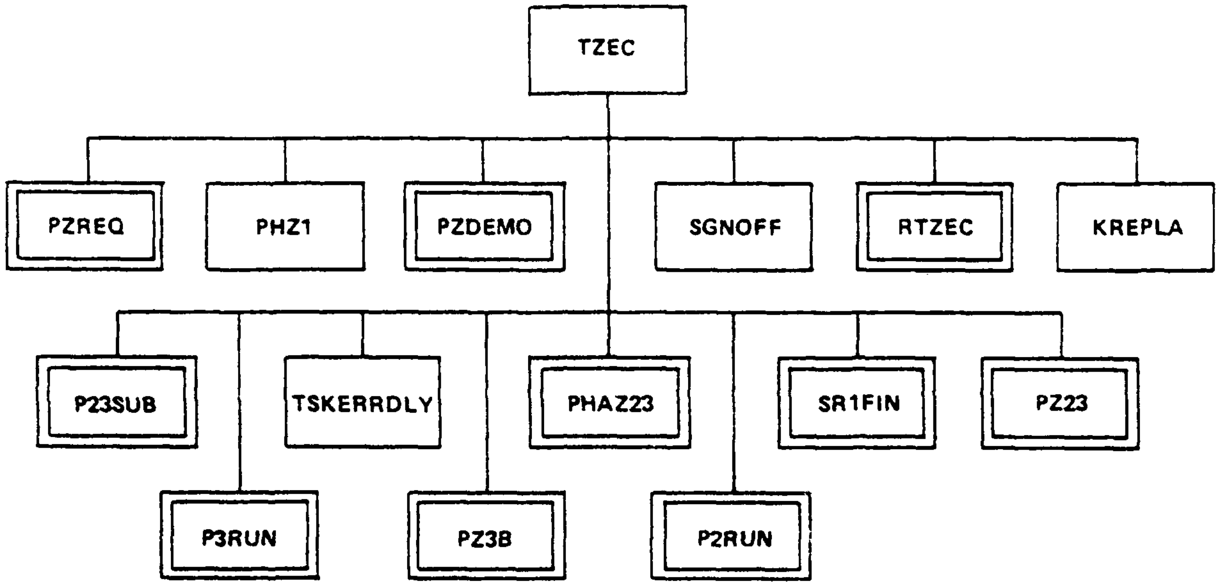


Figure 17. Block Diagram Overview of Training Control

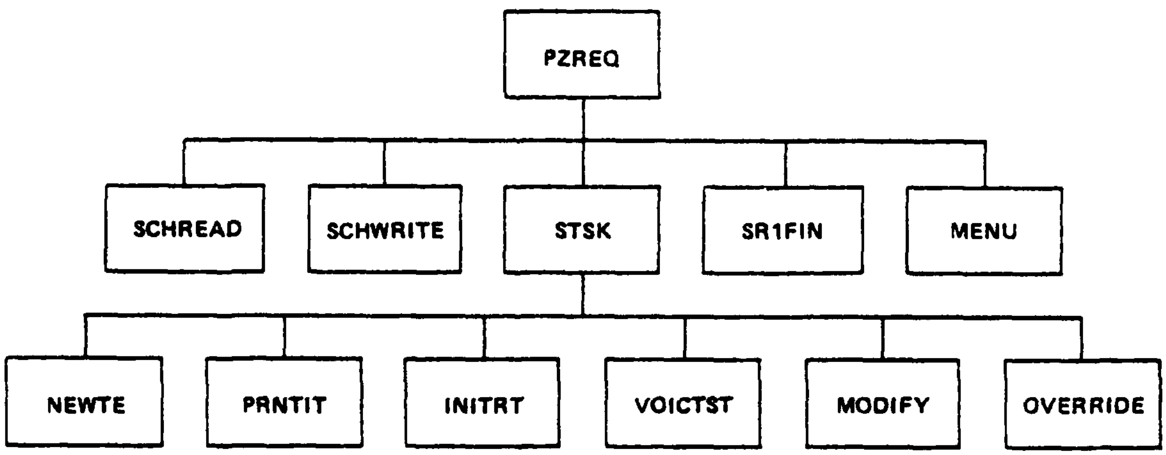


Figure 18. Block Diagram of Special Requests for Training Control

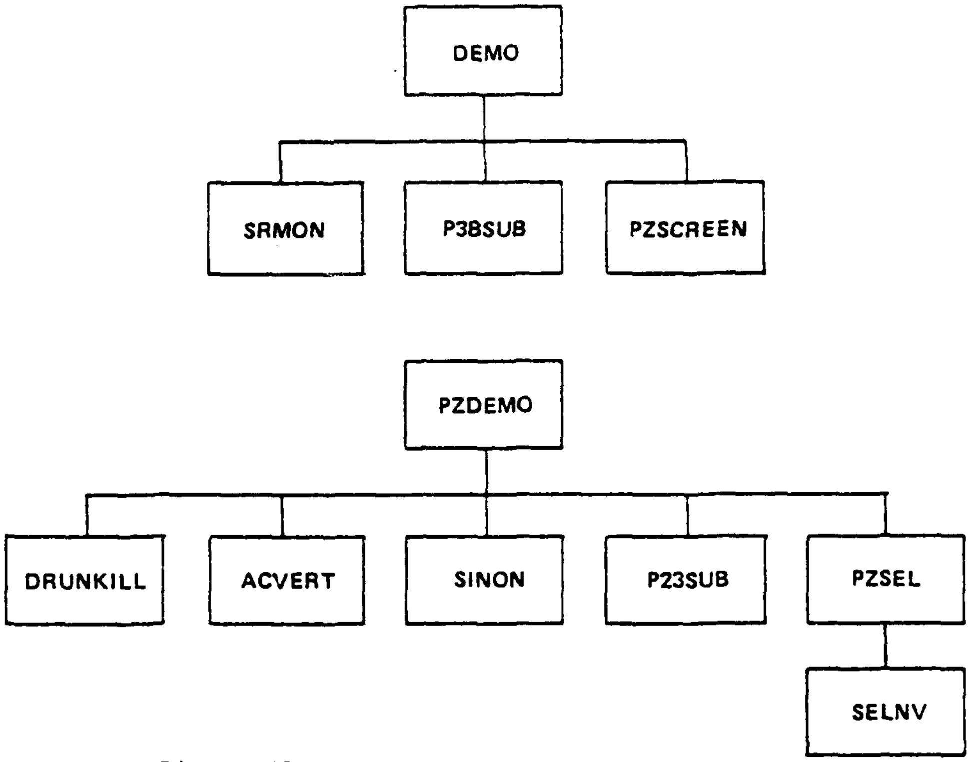


Figure 19. Demonstration Mode Block Diagram

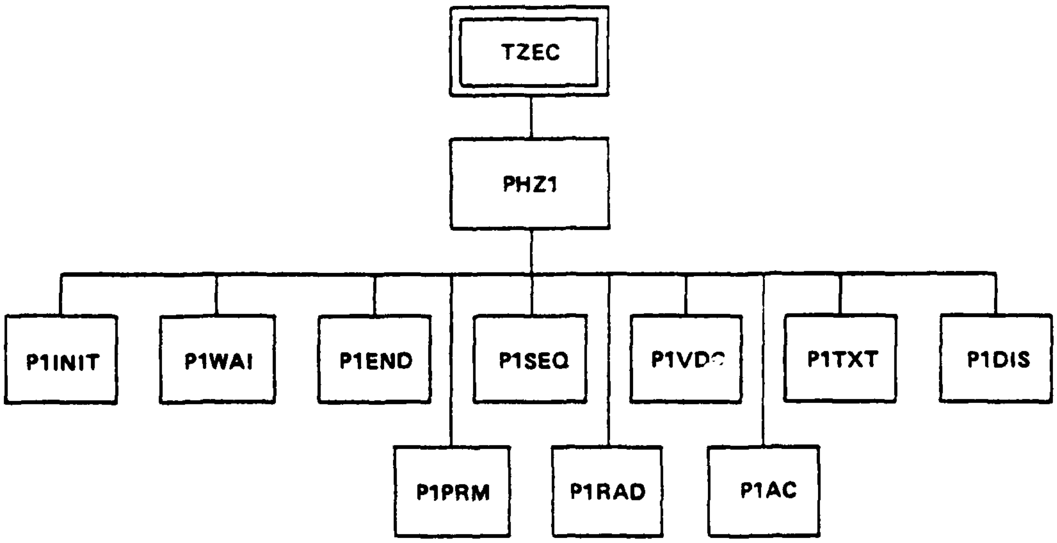


Figure 20. Overview Block Diagram, Phase 1

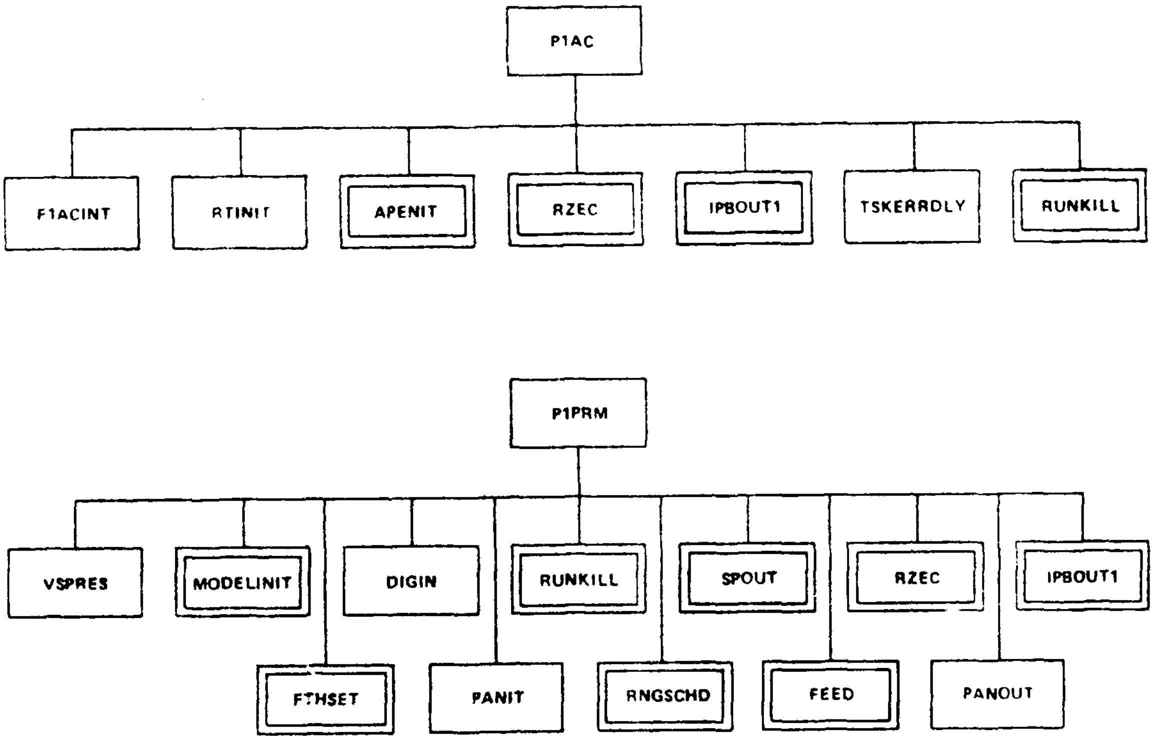


Figure 21. Phase 1 Run Initialization and Execution Block Diagrams

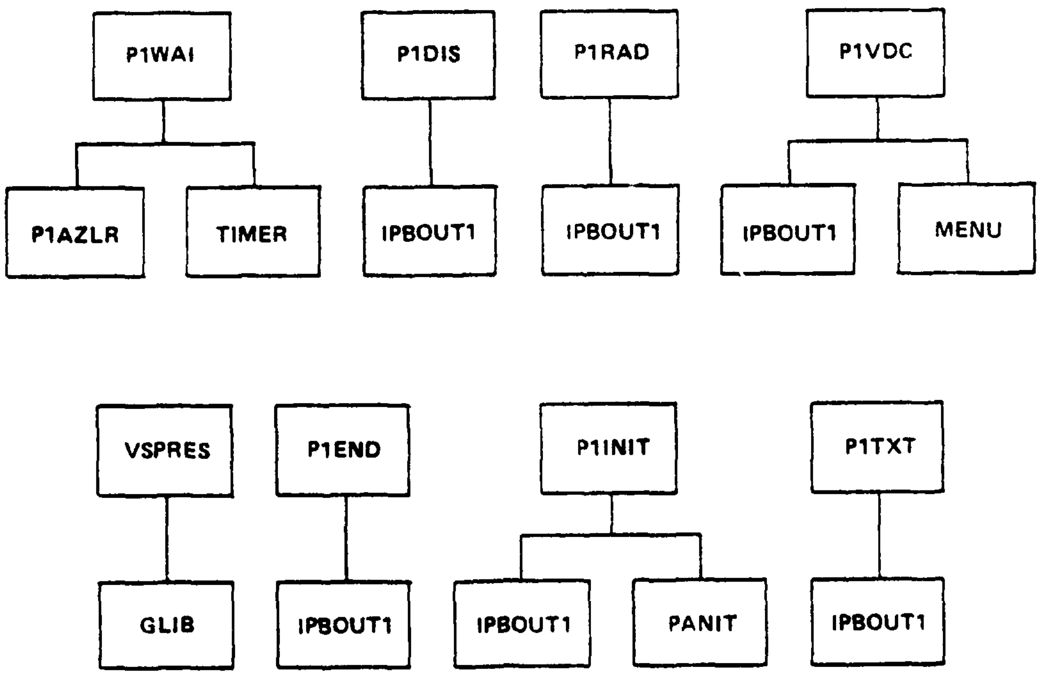


Figure 22. Other Phase 1 Block Diagrams

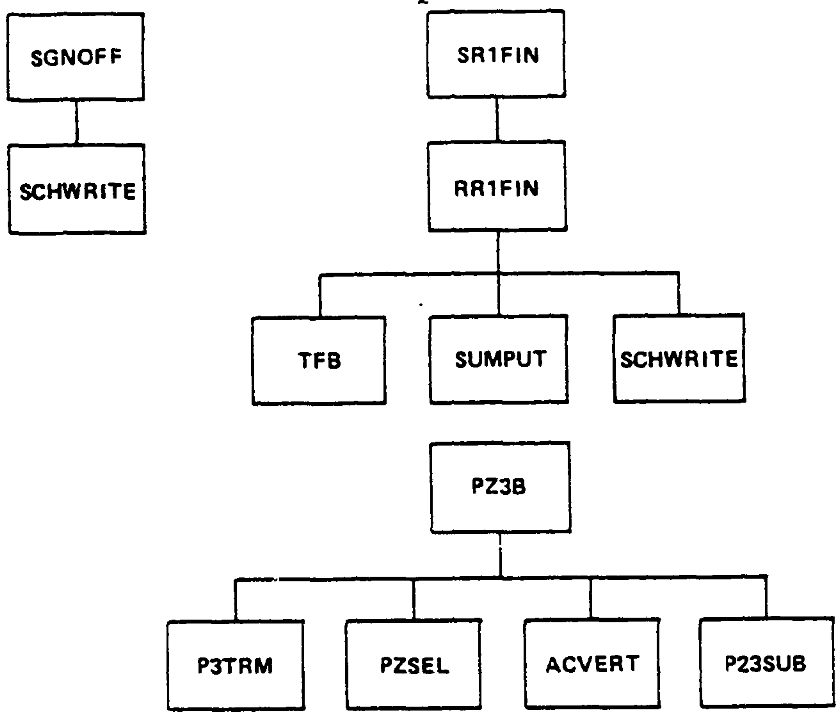


Figure 23. Block Diagrams for Multipossibility Executive and Student File Access.

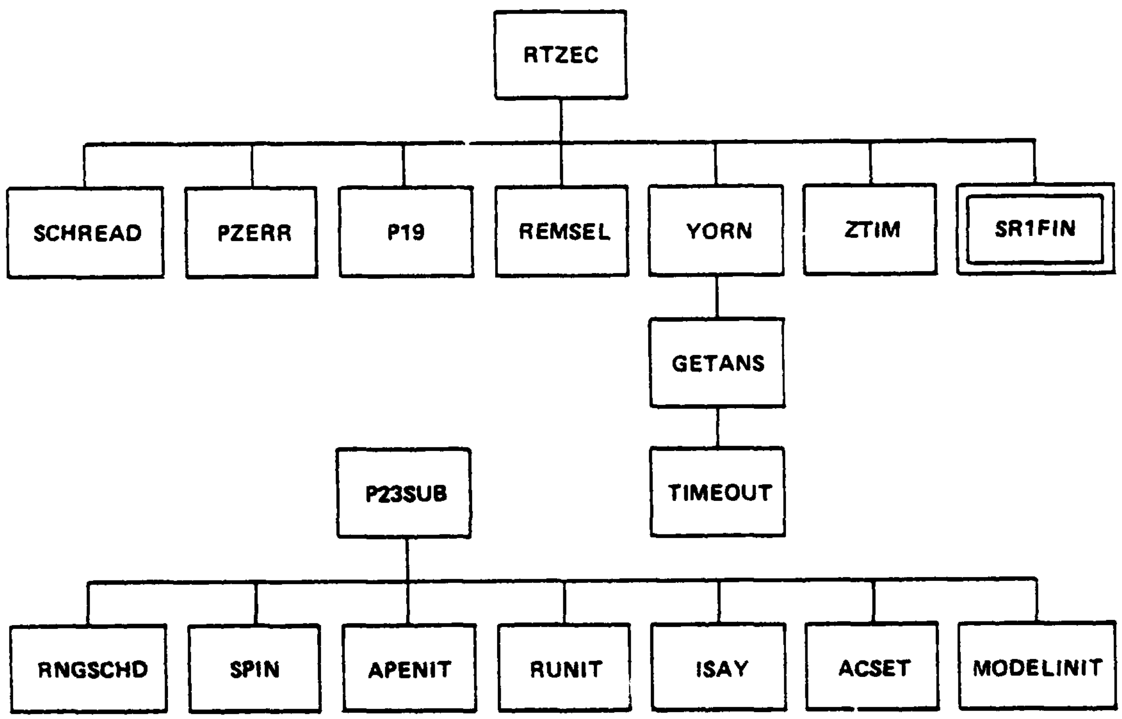


Figure 24. Block Diagrams for Pre-Run Initialization and Problem Selection

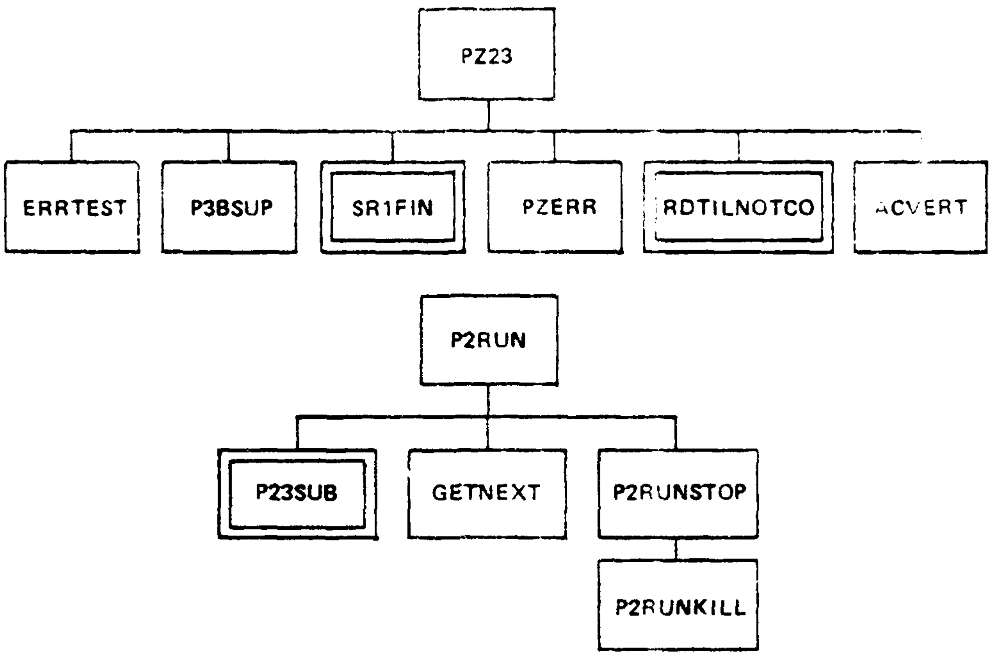


Figure 25. Block Diagrams for Single Possibility Card Reader and Phase 2 Executive

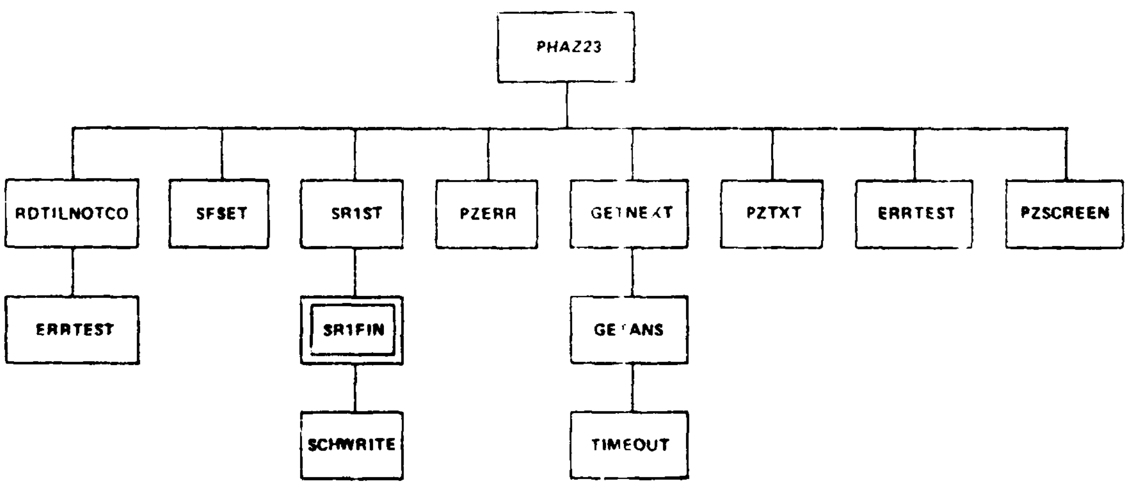


Figure 26. Block Diagrams for Phase 2 and 3 Routines Handling Header Cards for Single and Multipossibility Problems

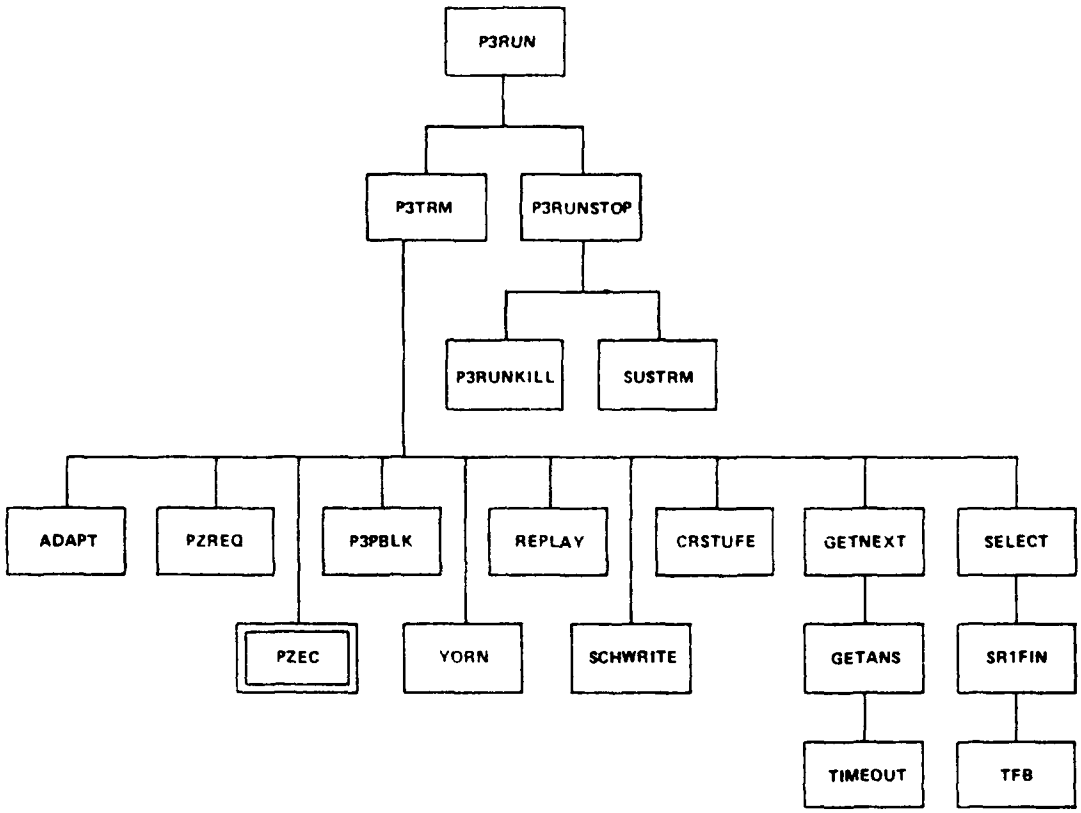


Figure 27. Block Diagrams for Phase 3 Executive

task by requesting an override via the keyboard. If the trainee finishes the standard course early, enrichment tasks are available. The training plan can be changed simply by monitoring the syllabus file or the task file.

The master routine to control the training process is called TZEC. This Fortran task always begins in the demonstration mode. Most routines are called by TZEC with the dummy argument NEXT. Upon return from the routine, NEXT is set to indicate the next appropriate task which TZEC, or one of TZEC's subordinate routines, is to perform. For example, if a trainee signs on, NEXT will be set to nine to commence training. Control is always returned to TZEC between phases. TZEC can also cause special requests from the instructor or trainee keyboard to be processed. These requests can initiate voice tests, override tasks or terminate the training session. The training system will also terminate itself if the trainee does not respond for a given length of time. TZEC also keeps track of the trainee's time on the system for the day, provides breaks and initiates termination if the trainee wishes to leave.

MODES OF OPERATION

PHASE 1 TRAINING EXECUTIVE. PHZ1, the phase 1 training executive, handles the processing of phase 1 task files. Phase 1 task commands initiate voice data collection, Megatek displays, terminal CRTs, and digitized voice prompts, and it waits on student responses, aircraft demonstrations, servo conditions, and task file sequencing logic. Explanations of command types and their associated instruction formats are included in the discussion on phase 1 task file structure in Appendix C.

Voice data collection is requested through subroutine P1VDC. P1VDC simply sends arguments to the SPEECH task on CPU 2 via the IPB.

Display instructions are also handled in a similar manner by P1DIS. The task IMAGES is the receiver on CPU 2 in this case.

Prompts, under the jurisdiction of P1PRM, are directed to the appropriate output device controller. The student CRT prompts are transferred via the IPB. The model controller prompt option also relies on the expertise of the EXPERT modules to select proper model controller advisories.

Radar servo conditions are initiated and frozen by P1RAD. Azimuth and elevation servos are positioned, aligned, activated, and deactivated via the IPB messages to SERVO on CPU 2. Servo alignment conditions are also initialized by P1RAD via a common block for the RADAR routine.

Aircraft approach simulation initial values are set by P1AC. The simulation is begun on request, at which time a task which freezes on an event parameter is activated.

A couple of wait conditions, handled by P1WAI, are also applicable to aircraft simulation events, such as, wait for aircraft to enter azimuth zone. Other types of general wait conditions are also provided.

Task file sequencing instructions provide skips, if statement constructs, and subroutine constructs. Abnormal subroutine returns also provide returns to an offset from the normal return point. Nested subroutines are permitted up to five levels.

All instructions are defined in card format to form a primitive interpretive training language. Any phase 1 task file may be easily modified by replacing, inserting, or removing cards with care given to preserving sequence instruction validity.

PHASE 2 TRAINING EXECUTIVE. Phase 2 is an optional practice mode offered to the trainee upon completion of a phase 1 task. Each phase 2 task may have several problems in it. There is a header card in each phase 2 file which indicates which behaviors are to be monitored by the phase 2 executive. In general, the behaviors monitored correspond to the performance variables to be scored for the related tasks, although such a condition is not absolutely necessary. A phase 2 run behaves like a normal approach until the trainee responds incorrectly to a flagged behavior. At this time the run freezes and an explanation of the error is given via the CRT. The run is then restarted. Each problem may be run several times. A problem automatically terminates if the trainee makes an error three times. To accomplish the flagging of errors, the performance measurement module is active. At least once each half second a record is stored in a buffer indicating the present state of the world. The performance module compares this state with the phrase recognized by speech recognition. If an error occurs, performance measurement signals the executive. If the trainee completes a phase 2 run successfully, the run can be repeated a variable number of times dependent on a value specified in the phase 2 problem file. Each phase 2 run is a single possibility run. This means that the parameters of the problem, including wind, type of approach, starting and ending range, speed of the aircraft, etc., are fixed for each problem in the file. Because phases 2 and 3 are similar in execution for single possibility problems, much of the logic for the initiation of a problem is collapsed into general routines. Thus PZ23 reads the single possibility cards and sets the common variables for phase 2 and phase 3. The routine P2RUN is called to control the phase 2 run. This routine determines the cause of termination, either a freeze on error or successful end of run, and selects the next task. If the cause of termination was a freeze on error, and not the third error on the problem, P2RUN forces the phase 2 problem file pointer back to the beginning of the current phase 2 run and returns control to TZEC, the training executive, to restart the problem. If not, P2RUN indicates that a new phase 2 problem is acceptable and again returns control to TZEC to proceed to the next problem.

PHASE 3 TRAINING EXECUTIVE. Phase 3 runs and performance runs are aircraft approaches, similar to those seen in a genuine controller station. These runs are always executed by the trainee and the conditions of flight closely approximate real world approach conditions. Phase 3 runs can be either single or multipossibililty. Performance runs, or P-runs are always single possibility. In a single possibility run, the conditions for an approach are fixed for each problem. In multipossibility problems, the header card for each problem file contains the percentage desired for each condition of a flight.

Thus one phase 3 problem file might require 50% straight-in approaches, 25% right and 25% left base. A routine called PGSEL receives cumulative percentages computed by P3BSUB and uses them to determine the conditions for each phase 3 problem. When the trainee gives an advisory, the pilot will respond, even if the advisory is incorrect. If a trainee performs poorly, the phase 3 routine ADAPT will flag the conditions that should be modified. Adaptation can be performed on aircraft speed, pilot ability and wind fluctuation. If a previously mastered skill is degrading, a remedial task can be assigned to the trainee by the routine SELECT. In either case, a notation of the adaptation or remediation is made in the trainee's file.

After a run is completed, the performance measurement executive is activated to score the run. All scores for the trainee are stored on disk for possible later retrieval.

REPLAY. After a phase 3 problem or the P-run, feedback is given to the trainee regarding his performance. At that time, the trainee can request a replay of the run in either of two modes: with or without error reporting. During both types of replay, the trainee's own voice is played back in synchrony with the aircraft dynamics, with pattern controller and pilot dialog, and with light displays on the trainee panel. In addition, during the errors-reported mode of replay, the system freezes at the point where an error was detected by PMS, explains the error to the trainee, and resumes replay when the student indicates that he is ready to proceed.

In the case of phase 3 problems (as distinct from P-run), P3TRM allows the trainee to request a replay of the most recent approach as many times as he chooses. However, once he has elected to advance to the next problem, the old replay files are destroyed, making replays of previous approaches impossible.

P-run replays are similar to phase 3 replays, with one difference being that the replay files are not destroyed by subsequent runs and thus a replay of the P-run is available at any time by the selection of the REPLA key at the instructor station. This keyboard replay selection is handled by the KREPLAY module. (In addition, P-run files can be modified by the instructor, who can then call for a re-scoring of the run. This will be discussed in the section on MODIFY.) Refer to Figure 28 for a diagram of REPLAY.

Type 1 REPLAY (Errors Not Reported). Data for use by REPLAY are saved during the approach and placed in three files described more fully in the appendices. These files consist of radar information (RPLDSP or RPPDSP), student activity (RPLACT or RPPACT), and digitized speech (RPLSPH or RPPSPH). The REPLAY module initiates the RDRPLY and ATRPLY tasks which maintain core buffers of radar and student activity data, respectively, and process those data on the same half-second periodic that obtains during the actual run.

Replay of the aircraft target display involves merely sending the aircraft position data, which has been saved by the RADAR module, to the display processing module in CPU 2. This method obviates any need to exercise the aircraft dynamics and pseudo-random environmental simulator during REPLAY.

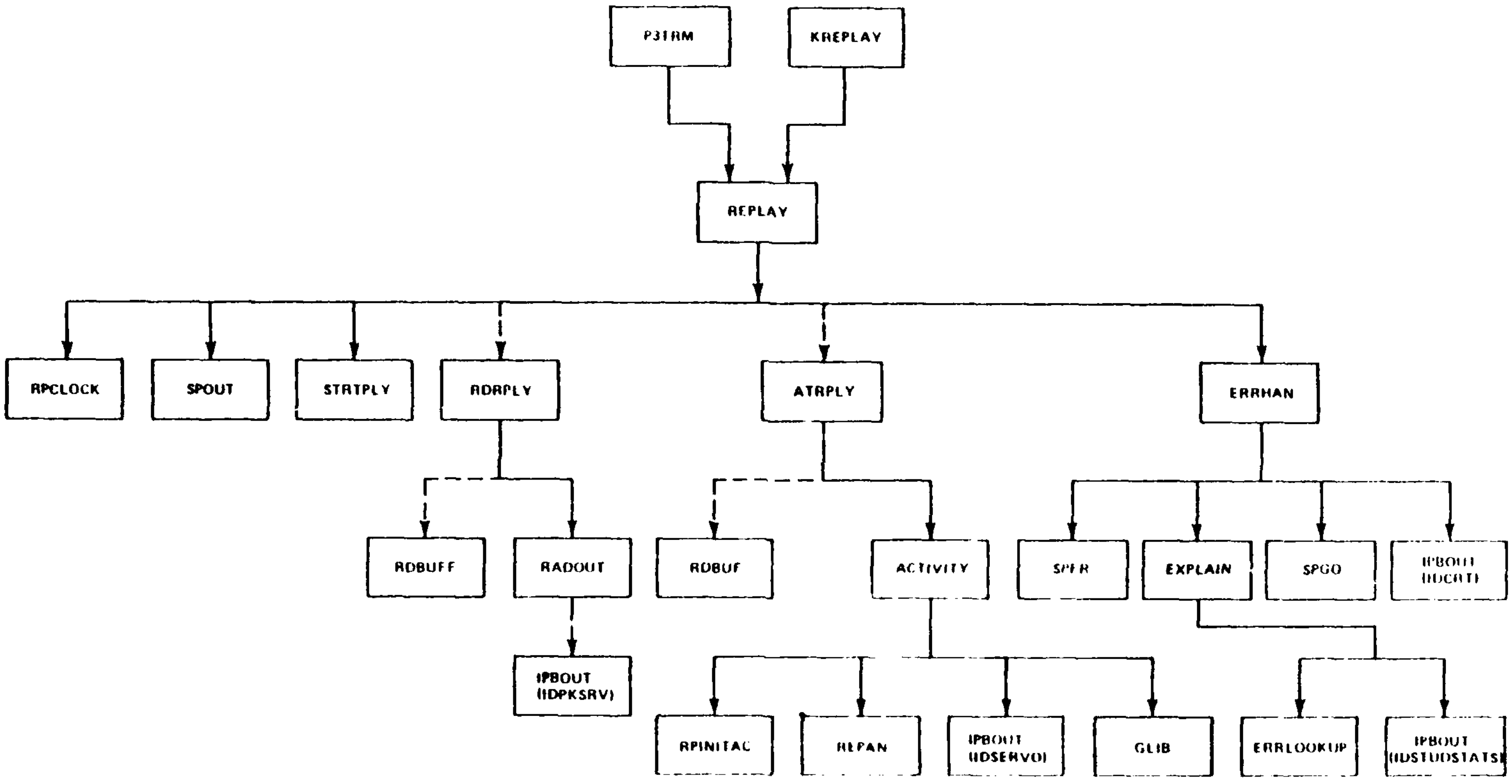


Figure 28. Structure of the Replay Module

The RDRPLY task calls RADOUT with a buffer full of radar display information, and RADOUT handles the timing considerations by comparing the time stored with each RPLDSP record with the time of the replay clock (RPCLOK). It waits until the time is right, sends the information to CPU 2, and goes on to the next record.

Replay of panel changes, servo position changes, and speech synthesizer output is marshalled by ARTPLY which, much like RDRPLY, maintains a buffer and calls ACTIVITY to go through that buffer and, at the correct time, to send information to various modules depending on the types of records it finds.

Finally, preparation for replay of the student's own voice is made by SPOUT and the playback is started by SPGO. Since the speech digitizer, unlike student activity and radar display, records continuously, it can be played back continuously without making any reference to the replay clock. Therefore, digitized voice is merely started and stopped at the appropriate time and its output rate is automatically controlled by the device itself.

Type 2 REPLAY (Errors Reported). The replay option which reports errors depends, additionally, on a fourth file created during the scoring of the run which contains errors and their times (ER or PER). In this mode, REPLAY suspends itself until it is time to report the next error. After this suspension it checks whether the speech digitizer is in the middle of a phrase. As soon as the digitizer is silent, REPLAY calls ERRHAN.

ERRHAN is designed to handle errors by freezing the replay (stopping the replay clock and speech digitizer), calling EXPLAIN to output the information to the student, waiting for the student to indicate that he is ready to continue, and restarting the replay.

EXPLAIN accepts an argument from the error file (via ERRHAN) which is a pointer into a file of error explanations (ERXFI). Each record in ERXFI contains some text to output and indicators as to whether any state-of-the-world information (e.g., present call sign, correct frequency, etc.) is needed. If such information is needed, ERLOOKUP is called with the indicator. ERLOOKUP acquires the information desired and outputs it.

DEMONSTRATION. The demonstration mode is very similar to the phase 3 run. Instead of input from the trainee, however, the final controller model executes the approach using the VOTRAX for speech output. A flag (CTON) is set to indicate that the final controller model is active. The demonstration mode is active whenever the GCA-CTS system is up and a trainee is not signed on. Demonstrations can also be performed during phase 1. Although at this time all demonstration problems are multipossibility, the facility is available to have single possibility demonstrations. Multipossibility problems were implemented to allow a large number of approaches to be executed between trainees. When a demonstration is in progress, the activity file is not filled.

SOFTWARE RELATED TO SPEECH UNDERSTANDING

This set of routines is the subsystem which allows verbal input to GCA-CTS. Speech understanding consists of both recognizing phrases and transforming phrases into messages. This introduces the distinction between phrases and messages. In this context, a phrase is a group of one or more words without an internal pause. A complete GCA-CTS phrase list appears in Table 8. A message is a complete GCA message. A list of GCA messages appears in Table 9. (Note that some but not all phrases constitute messages. This will be discussed again in the Speech Understanding section.

PHRASE TRAINING AND RECOGNITION. This part of the speech understanding system consists of creating recognition patterns for phrases (training the machine) and matching input patterns with stored patterns (recognizing phrases). This section is divided into two parts to reflect this structure.

VOICE DATA COLLECTION. Voice data collection (VDC) and validation is inherent to phase 1 instruction. While the student is taught the correct radio terminology, he is encouraged to repeat and practice voicing the phrases. During this process, voice input feature patterns may be easily obtained without requesting unnatural repetition of terminology. Refer to Figure 29 for the structure of this set of routines.

The routines VDCON and VDCOFF are required to prepare for VDC and to terminate VDC. Channels are opened and closed to the voice data files. After VDC is turned on, the routines COLLECT and FORMIT may be used to COLLECT input feature patterns and to FORMIT into voice reference patterns.

Input Collection. COLLECT does not issue prompts automatically but does re-prompt if a faulty input feature pattern is detected: incorrect pauses, incorrect concatenations or a long drawn-out phrase. It does so by counting phrases input and checking for buffer overflow. The buffer is examined for leftover phrases input after the correct number is collected.

The features are input via the Threshold 500 driver (VIPDR) to be processed by input feature pattern formulation routine VIFP.

Input Feature Patterns (IFPs). Every 2 milliseconds (approximately) the Threshold 500 preprocessor generates an interrupt and provides two 16-bit words via the interface logic. Each of these 32 bits corresponds to a feature. A bit will be set (i.e., equal to a one) if, and only if, the corresponding feature was present in the voice sample.

All of these data are stored in memory by an interrupt service routine (VIPDR). One of the features (LP_4) indicates a long pause. When this feature is set, the system assumes that the phrase is complete and stops accepting data from the preprocessor.

The system now initiates a time normalization process. That is, regardless of the length of the phrase, all of the data in the input buffer are squeezed into an input feature pattern (IFP) which has only 16 or 32 time

TABLE 8. GCA-CTS PHRASES

Number	Number of Repeats	Phrase Identifier	Phrase
1)	4	004302	1 MILE
2)	4	104303	1 AND 1/2 MILES
3)	4	004304	2 MILES
4)	4	104305	2 AND 1/2 MILES
5)	4	004306	3 MILES
6)	4	104307	3 AND 1/2 MILES
7)	4	010101	AT
8)	4	050105	TOWER CLEARANCE CANCELLED
9)	4	050106	TOWER CLEARANCE NOT RECEIVED
10)	-		NOT USED
11)	-		NOT USED
12)	-		NOT USED
13)	10	034500	0
14)	10	034501	1
15)	10	034502	2
16)	10	034503	3
17)	10	034504	4
18)	10	034505	5
19)	10	034506	6
20)	10	034507	7
21)	10	034510	8
22)	10	034511	9
23)	4	144021	CONTACT TOWER AFTER LANDING
24)	4	144024	BUTTON 1 CLEAR
25)	4	144026	BUTTON 2 CLEAR
26)	10	044100	MISSED APPROACH
27)	4	144101	IF RUNWAY NOT IN SIGHT
28)	4	144102	IF RUNWAY NOT IN SIGHT EXECUTE MISSED APPROACH
29)	-		NOT USED
30)	4	044104	BUTTON 1
31)	4	144105	PROCEED DIRECT POINT BRAVO HOLD UNTIL ADVISED BY GCA
32)	4	044106	BUTTON 2
33)	4	044110	ON THE GO
34)	4	144220	OVER LANDING THRESHOLD
35)	4	145000	TOO FAR LEFT FOR SAFE APPROACH
36)	4	145010	TOO FAR RIGHT FOR SAFE APPROACH
37)	4	145220	ON CENTERLINE
38)	4	145222	LEFT OF CENTERLINE
39)	4	145224	SLIGHTLY LEFT OF CENTERLINE
40)	4	145232	RIGHT OF CENTERLINE
41)	4	145234	SLIGHTLY RIGHT OF CENTERLINE
42)	4	146001	TOO LOW FOR SAFE APPROACH
43)	4	146011	TOO HIGH FOR SAFE APPROACH
44)	10	050100	WIND

TABLE 8. GCA-CTS PHRASES (CONT)

Number	Number of Repeats	Phrase Identi-fier	Phrase
45)	4	150102	CLEARED FOR LOW APPROACH
46)	4	150103	CLEARED FOR TOUCH AND GO
47)	4	050104	CLEARED TO LAND
48)	4	150201	1 MILE FROM TOUCHDOWN
49)	4	150202	2 MILES FROM TOUCHDOWN
50)	4	150203	3 MILES FROM TOUCHDOWN
51)	4	150204	4 MILES FROM TOUCHDOWN
52)	10	151001	WELL LEFT OF COURSE
53)	10	051002	LEFT OF COURSE
54)	10	151011	WELL RIGHT OF COURSE
55)	10	051012	RIGHT OF COURSE
56)	10	152001	WELL BELOW GLIDEPATH
57)	10	152011	WELL ABOVE GLIDEPATH
58)	10	152023	GOING FURTHER BELOW GLIDEPATH
59)	10	152033	GOING FURTHER ABOVE GLIDEPATH
60)	4	154103	CLIMB AND MAINTAIN 1500
61)	4	154200	AT DECISION HEIGHT
62)	10	055000	ON COURSE
63)	10	155004	SLIGHTLY LEFT OF COURSE
64)	10	155014	SLIGHTLY RIGHT OF COURSE
65)	10	055033	CORRECTING
66)	10	056000	ON GLIDEPATH
67)	10	156002	BELOW GLIDEPATH
68)	10	156004	SLIGHTLY BELOW GLIDEPATH
69)	10	156012	ABOVE GLIDEPATH
70)	10	156014	SLIGHTLY ABOVE GLIDEPATH
71)	10	156024	GOING BELOW GLIDEPATH
72)	10	056027	COMING UP
73)	10	156034	GOING ABOVE GLIDEPATH
74)	10	056037	COMING DOWN
75)	4	160001	POSITION 4 ROGER
76)	4	160004	RADAR BUTTON 1
77)	4	160006	RADAR BUTTON 2
78)	4	160007	THIS IS YOUR FINAL CONTROLLER, HOW DO YOU HEAR ME?
79)	4	160010	WHEELS SHOULD BE DOWN
80)	4	160011	DO NOT ACKNOWLEDGE FURTHER TRANSMISSIONS
81)	10	160012	APPROACHING GLIDEPATH
82)	4	160013	BEGIN DESCENT
83)	4	160014	GIVE ME BUTTON 1
84)	4	160016	GIVE ME BUTTON 2
85)	4	164001	ARMY 876
86)	4	164002	MARINE 687
87)	4	164003	NAVY 310
88)	4	164004	AIR FORCE 307
89)	4	064010	OVER

TABLE 8. GCA-CTS PHRASES (CONT)

Number	Number of Repeats	Phrase Identi- fier	Phrase
90)	4	170040	THIS WILL BE A NO GYRO PAR APPROACH
91)	4	170041	MAKE HALF STANDARD RATE TURNS
92)	4	170205	5 MILES FROM TOUCHDOWN
93)	4	170206	6 MILES FROM TOUCHDOWN
94)	4	170207	7 MILES FROM TOUCHDOWN
95)	4	170210	8 MILES FROM TOUCHDOWN
96)	4	172001	LOW ALTITUDE ALERT CHECK YOUR ALTITUDE IMMEDIATELY
97)	4	174000	HOW DO YOU HEAR ME NOW?
98)	10	074000	CORRECTION
99)	10	074043	TURN RIGHT
100)	10	074047	STOP TURN
101)	10	074053	TURN LEFT
102)	10	174100	EXECUTE MISSED APPROACH
103)	10	174101	RADAR CONTACT LOST
104)	4	174102	CLIMB AND MAINTAIN 3000
105)	10	174403	TURN RIGHT HEADING
106)	10	074407	HEADING
107)	10	174413	TURN LEFT HEADING
108)		000001	NOT RECOGNIZED
109)		000002	TOO SHORT
110)		000003	TOO LONG
111)		000004	TOO SOFT

TABLE 9. GCA MESSAGES

	Message
1)	"Position X, Roger" (X = numeral, single digit)
2)	"C/S, Radar Button X" (C/S = call sign of aircraft, e.g., Navy; XXX = numeral, 3 digits)
3)	"Give me Button X"
4)	"C/S, this is your final controller how do you hear me?"
5)	"How do you hear me now?"
6)	"C/S, turn right heading XXX, over"
7)	"C/S, turn left heading XXX, over"
8)	"Wheels should be down, over"
9)	"On the go, C/S, Button X."
10)	"On glidepath"
11)	"Above glidepath"
12)	"Below glidepath"
13)	"Slightly above glidepath"
14)	"Slightly below glidepath"
15)	"Well above glidepath"
16)	"Well below glidepath"
17)	"Coming up"
18)	"Coming down"
19)	"Going further above glidepath"
20)	"Going further below glidepath"
21)	"Going above glidepath"
22)	"Going below glidepath"
23)	"Tower clearance cancelled, execute missed approach, [climb and maintain 1500, turn right heading 300]"

TABLE 9. GCA MESSAGES (CONT)

Message

24) "Tower clearance not received, execute missed approach, [climb and maintain 1500, turn right heading 300]"

25) "Heading XXX"

26) "C/S, approaching glidepath, over"

27) "Approaching glidepath"

28) "Begin descent"

29) "Missed approach, C/S, (map position), Button X"

30) "Well right of course, turn left heading XXX"

31) "Well left of course, turn right heading XXX"

32) "Well right of course, correcting"

33) "Well left of course, correcting"

34) "Slightly left of course"

35) "Slightly right of course"

36) "On course"

37) "X mile(s) from touchdown"

38) "At decision height"

39) "At decision height, too X for safe approach, if runway not in sight, execute missed approach, [climb and maintain 1500, turn right heading 300]"

X = 1) high
2) low
3) far right
4) far left

40) "Wind XXX at X, cleared X_1"

X_1 = 1) for low approach
2) for touch and go
3) to land

41) "C/S, do not acknowledge further transmissions"

TABLE 9. GCA MESSAGES (CONT)

Message

42) "Over landing threshold, [X centerline], over"

X = 1) left of
2) slightly left of
3) on
4) slightly right of
5) right of

43) "Over"

44) "Contact tower after landing, over"

45) "Button X, clear"

46) "Radar contact lost, [if runway not in sight] execute missed approach, climb and maintain 3000, turn right, proceed direct point bravo hold until advised by GCA"

47) "Low altitude alert check your altitude immediately"

48) "C/S, this will be a no-gyro PAR approach"

49) "C/S, turn left, over"

50) "C/S, turn right, over"

51) "C/S, stop turn, over"

52) "This will be a no-gyro PAR approach"

53) "Make half standard rate turns"

54) "Turn left"

55) "Turn right"

56) "Stop turn"

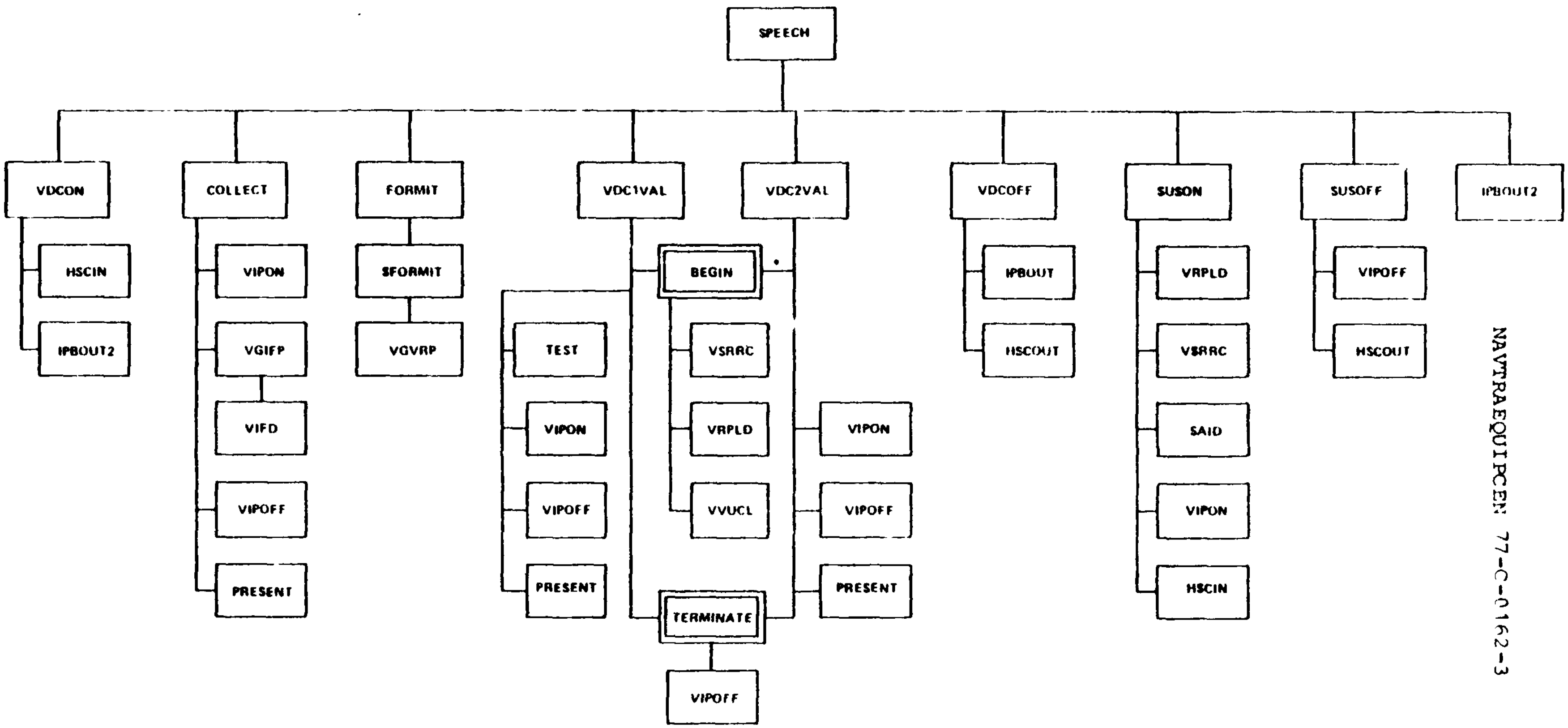


Figure 29. The Structure of Voice Data Collection

slots. This is done by dividing the samples in the buffer into 16 or 32 segments. If a feature is set for a quarter or more of the samples in each segment, that feature is set in the corresponding time slot of the IFP. Since the optimum number of time slots is related to the length of the input utterance, 16 time slot patterns are stored for short phrases (3 syllables or less). For all other phrases, 32 time slots are stored.

Reference Pattern Formulation. FORMIT is the routine which is responsible for voice reference pattern creation. A test is made to assure the required number of IFPs is available. COLLECT and FORMIT are normally requested by phase 1, but may be accessed via instructor keyboard entries if a faulty voice reference pattern is suspected. New inputs are used to update the outdated patterns.

Voice Reference Patterns (VRPs). The VRP is formulated from the IFPs using the repetition count (number of IFPs stored).

To understand the problems in this area, a closer look at the implications and assumptions surrounding the repetitive voicing of vocabulary phrases is required. Each item of the vocabulary is repeated in order to:

a. Pick up all features that are characteristic of the student's voicing of a phrase - but, at the same time,

b. Exclude features that are only sometimes present and are not characteristic of the phrase itself.

To achieve this, Threshold Technology has suggested that for M repeats, a feature/time element must be present in N IFPs in order to be present in the VRP. Table 10 shows the relationship between M and N for one to ten repeats.

TABLE 10. RELATIONSHIP BETWEEN NUMBER OF REPEATS AND IFP BIT SETTINGS

Number of Repeats M	Number of IFPs N	
1	1	
2	1	Zone 1
3	1	
4	2	
5	2	Zone 2
6	2	
7	3	
8	3	Zone 3
9	3	
10	4	Zone 4

Generally speaking, the repetition count can be considered in zones: 1-3 repetitions in the first zone, 4-6 in the second, 7-9 in the third, and 10 in the fourth. The second objective is furthered by moving from zone to zone. The first objective is furthered by increasing the repetition count within a zone.

Notice that the first three repetitions will put all features into the VRP. Thus no contribution is made toward the second objective previously cited.

With the fourth repetition, the number of features put into the VRP is reduced considerably, but the number starts to build up again until the seventh repetition, and so on. The number of features set in the VRP is therefore related - in a very nonlinear way - to the number of repetitions.

The score that will be calculated during the speech recognition process by correlating an IFP with each VRP is a function of the number of features set in the VRP (and IFP). So, to compare scores, it is necessary to ensure that all VRPs have been formed with the same probability of success.

Past tests have produced a correlation between score and repetition count. Therefore, the following conclusions resulted:

a. No phrase should be trained less than four times.

b. Phrases which are easily confused (e.g., "above glidepath" and "below glidepath") should be trained an equal number of times.

In implementation, then, four IFPs are collected for the phrases which are not likely to be mistaken for another phrase. For other phrases which are likely to produce confusion, ten IFPs are collected before a voice reference pattern is formulated.

Validation. The purpose of this mode of VDC program operation is to further instruct a student and/or allow him to practice with the radio terminology. Familiarization with the speech recognition system is a beneficial side effect.

The validation mode actually consists of two submodes, VDCVAL1 and VDCVAL2. Using VDCVAL2, the system will not prompt the student; rather, the student prompts the system. That is, the system will attempt to echo the phrases which it recognizes the student to be speaking. This presentation occurs one second after the student is silent.

In the VDCVAL1 mode, the program will prompt the student in the same way as in the training mode, utilizing the latest presentation device. Recognition accuracy is recorded in the voice data file. When the requested recognition accuracy has been achieved for three consecutive prompts, the validation mode is terminated.

Subroutines BEGIN and TERMINATE apply to both validation modes. They perform the functions that their names imply. PRESENT serves both VDCVAL1 and VDCVAL2 as a prompting and mimicking routine, respectively.

SPEECH RECOGNITION. Speech recognition (SR) compares an input phrase with stored VRPs to determine the identity of the phrases spoken. The identified phrases are then shipped to CPU 1 for speech understanding processing or are identified to the validation module. In the first case, SR is activated by SUSON and in the latter by BEGIN.

Phrases involved in the recognition process and the recognition logic details follow.

Phrases for Recognition. All final controller phrases have been classified and categorized. The purpose of this classification is to aid the speech recognition module in its search for a likely voice reference pattern (VRP) as well as to provide a logical grouping for the phrases.

The classification of phrases produces an identification word which describes the phrase content and use. By applying a masking scheme on the identification word, particular types of phrases may be singled out. In general the recognition algorithm examines the phrases which are most likely to have been uttered (valid phrase types are provided by the final controller modules) and then proceeds to examine the remainder of the phrases if a good recognition is not produced.

All interprogram communication with the voice recognition, speech understanding, and voice data collection modules reference phrases either by the sequence order or by the phrase identification. The sequence number is in decimal notation and reflects the order of the provided list. The identification word is expressed in octal notation.

Identification Words. The identification words are the result of bit settings which correspond to the phrase content and use. Each phrase is assigned 1 16-bit word with bits delegated as shown in Table 11 and as discussed below.

a. Bit 0, Number of time slots for VRPs, is a storage reference since all phrases with three syllables or less shall be stored in 16 time slots instead of the traditional 32. This bit is set to indicate 32 time slots.

b. Bit 1, Key phrases, are those phrases which can stand alone. For example, digits are not key phrases since they must be used in conjunction with headings or a wind message. This bit is set for a key phrase.

c. Bit 2, Approach phase 1, refers to the beginning of the GCA in which the handoff to the final controller is made. The final controller speaks to either the pilot or the pattern controller during parts of this phase. The bit is set if the phrase is valid during this phase. (Note that "approach phase" is distinct from the training phases of the GCA-CTS.)

TABLE 11. SPEECH RECOGNITION IDENTIFICATION WORD BITS

Bit	Meaning if Bit is Set
0	VRP has 32 time slots (if 0, 16 time slots)
1	Key Phrase
2	Phrase used in approach phase 1
3	Phrase used in approach phase 2
4	Phrase used in approach phase 3
5	Glidepath phrase
6	Course phrase
7	Heading phrase
8	Range phrase
9	Missed approach, waveoff, impending missed aproach, wind or clearance phrase
10	No-gyro phrase
11	Trend phrase or landing threshold phrase
12	Phrase implying aircraft is above or right (if 0, below or left)
13	Phrase implying aircraft is in "slightly" zone
14	Phrase implying aircraft is in zone ($\pm$)2
15	Phrase implying aircraft is in "well" zone (if 0, in "on" zone)

d. Bit 3, Approach phase 2, references the part of the approach which follows the initial handoff. It is defined to begin after the "begin descent" advisory. This bit is set if the phrase is valid during phase 2.

e. Bit 4, Approach phase 3, begins as decision height, radar contact lost, or a missed approach execution or waveoff is announced. Such phrases which accomplish this transition are also classified as legal in the phase which they terminate. For example, "at decision height" is classified as a phase 2 and 3 phrase. Phase 3 continues until the final controller has completed the final handoff.

f. Bit 5, Glidepath, is set for all glidepath-related messages. This also applies to "too low for safe approach," etc.

g. Bit 6, Course, is set for all course-related messages.

h. Bit 7, Heading, is set for all heading messages or parts of heading messages, namely the digits.

i. Bit 8, Range, is set for all range-related messages, e.g., "X miles from touchdown."

j. Bit 9, Missed approach/wind, is set for all phrases associated with a missed approach or a wind or clearance message. All types of waveoffs, "radar contact lost", and others which imply a pending waveoff are included.

k. Bit 10, No-gyro, is set for no-gyro advisories, no-gyro type turns inclusive.

l. Bit 11, Trend/land, is set for either glidepath or course trend messages or advisories which are associated with an aircraft's landing, e.g., "contact tower after landing."

m. Bits 12 through 15 are reserved for individual phrase identification within categories or for glidepath, course and heading messages. Individual phrase ID values are assigned to differentiate between phrases with the same traits expressed in bits 1 through 11. This is not the case, however, for glidepath, course, and their associated trend messages and heading messages. For glidepath, course, and heading messages, bits are assigned in the following manner:

(1) Bit 12 is set for:

(a) Glidepath messages which apply when the aircraft's center is above the glidepath.

(b) Course messages which apply when the aircraft's center is to right of course, including turn left, since turn left is associated with being to the right of course.

(c) Trend messages used while aircraft satisfies the "above glidepath" or "right-of-course" condition.

(2) Bit 13 is set for:

(a) Being slightly above/below glidepath.

(b) Being slightly right/left of course corresponding trends.

(c) Corresponding trends.

(3) Bit 14 is set for:

(a) Being between slightly and well, e.g., "below glidepath," "right of course."

(b) Its use for trends is valid for the same conditions.

(4) Bit 15 is set for:

(a) Being well above, below, right, or left.

(b) The same for trend validity.

(5) Example for bits 12-15: "Going further above glidepath" is used when:

(a) Bit 12 = 1, the aircraft is above the glidepath.

(b) Bit 13 = 0, the aircraft is past the slightly zone.

(c) Bit 14 = 1, the aircraft is transitting to well above zone.

(d) Bit 15 = 1, the aircraft is going through the well zone.

Phrase List. The GCA-CTS phrases were shown in Table 8.

Additional identification words are provided for unrecognizable phrases or low input level.

000001	Message not understood
000002	Message too short
000003	Message too long
000004	Input level low

Speech Recognition Logic. The speech recognition logic proceeds in a fairly sequential manner. When the speech recognition routine VSRRC is activated, it awaits notification of a voice input from the Threshold 500 driver, VIPDR. The VIPDR fills an available input buffer and sends a communication packet to VSRRC. Figure 30 displays the SR structure.

SR maintains two distinct input buffers, A and B. Since a sample consists of 32 bits of information and Eclipse words hold 16 bits of information, each of these identical buffers consists of two parts: part 1 holds the first 16 bits of a sample and part 2 holds the last 16 bits, as shown in Figure 31.

The size of each buffer, BFSZ, will be a function of the length of the longest phrase to be recognized. Since a set of 32 features (a sample) is provided every 2 milliseconds, the length of each part of the buffer should be N/2 words long, where N is the number of milliseconds in the longest phrase. This value is an assembly parameter.

Two distinct buffers are defined in order to accept speech data (features) from the preprocessor even though the processing on the previous phrase has not yet been completed. By double buffering these raw data, there will be only a slight chance of missing or losing a phrase or portion of a phrase.

Information on the location and nature of the raw feature data is passed from the input device driver via a communication packet. This packet contains pointers to the start of the two parts of the input buffer, a pointer to the last sample in the first buffer, and a pointer to a buffer use flag. It is the responsibility of VSRRC to clear this flag when the buffer is free for future use.

Another word in the communications packet is reserved for unrecognizable phrase error returns for instances such as buffer overflows or phrases which are too short. The LP_4 time in half second clock ticks and 100 millisecond increment offsets is also sent via the communications packet.

This subroutine VSRRC awaits notification of a voice input and forms IFPs if an error return is not received. A 32-time-slot IFP is formed as well as a 16-time-slot IFP. These are stored in the input buffer area. Subroutine VOVEX then handles the remainder of the recognition process.

VOVEX clears a scoring area in the input buffer to accommodate scores for all VRPs. VRPRT is then called upon to find a set of plausible VRPs. VCOMP performs the comparisons and scores the VRPs in relation to the IFP. If a high-scoring VRP is not found using the first set provided by VRPPT, a second and even a third pass is made until a matching VRP has been found. The highest scoring VRP and sometimes a close second choice are found by subroutine VCHOS. VALYZ re-examines, if necessary, the first and second choice recognitions and assigns a confidence indicator to the recognition. VSPCL then performs application-specific processing.

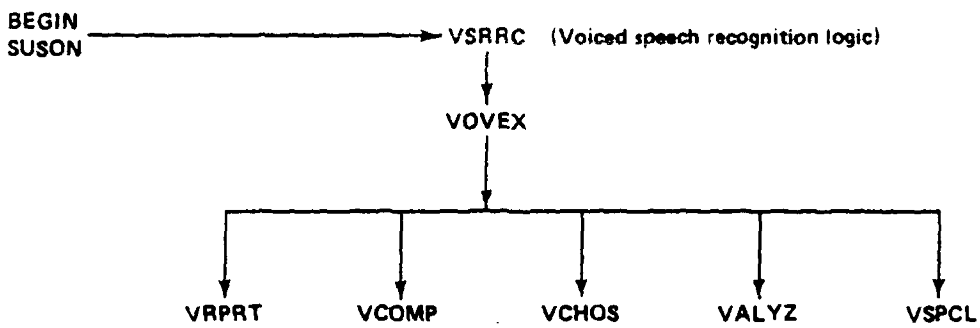


Figure 30. Voiced Speech Recognition Routines

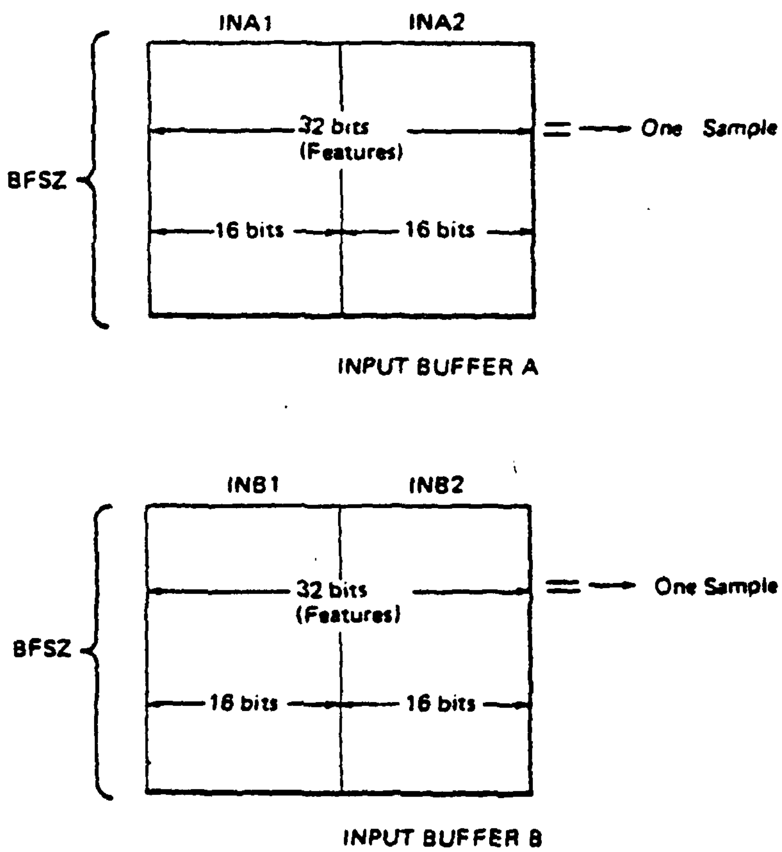


Figure 31. Speech Recognition Input Buffer Structure

VRPRT uses validation of SUS-provided phrase masks to locate sets of likely VRPs for recognition purposes. During validation the masks are set to accept all VRPs. The SUS mode requires masks provided by the phrase set chosen by the model controller. These masks include messages related to glidepath position (RCGPP), glidepath trend (RCGPT), course position (RCCRP), course trend (RCCPT), range (RCRNG), emergency (RCEMERG), and others (RCOTHR). Since the masks are set to appropriate final controller advisories, a first pass attempt made by VRPRT includes only phrases which are applicable to the correct glidepath position (above/below) or the correct course position (right/left) and likewise for the corresponding trends. Therefore, a low confidence conflict may not result between above/below or left/right. If none of the phrases in this first pass qualifies as a recognition in VCOMP and VCHOS, a second attempt is made using VRPRT with only the phase of flight as a mask. If this fails again, a third and final pass finds all remaining phrases yet untried.

When VRPRT finds a phrase whose identification word matches a mask, it first checks to see if a VRP exists for the phrase. If no VRP exists, the score is set to one. The two's complement of the phrase VRP record number is set in the score word of the phrase if a VRP does exist and a score has not already been computed. One complete pass through the identification words is made each time VRPRT is called.

VCOMP scans the score table for negative entries. The record number is translated to a block and word offset. This information is used to map in the necessary block which holds the VRP into the window buffer before the IFP/VRP comparison is made.

Since the VRPs are loaded into virtual memory when SUS is turned on (SUSCN), a window map scheme is necessary to access the VRPs. For this purpose a 1024-word buffer, VRPBF, has been defined as the window.

In VCOMP, the IFP is compared to each of the VRPs. This is done on a bit-by-bit basis, by the TTI-provided high speed correlator (HSC) using the algorithm given in Table 12. The number (score) for each bit-by-bit comparison of the IFP and VRP is totaled and saved for each VRP. This calculation is repeated for each IFP/VRP pair, but with IFP first shifted up by one time slot and then down by one time slot; the largest of the three scores is saved. All of these scores are normalized by the highest possible score, that is, the score that is obtainable if the IFP matched all VRP bits. If a negative score is obtained, it is set to 1 in the score table.

TABLE 12. IFP/VRP COMPARISON ALGORITHM

	Feature Set in VRP	Feature Not Set in VRP
Feature Set in IFP	2	-1
Feature Not Set in IFP	-1	0

The selection logic in VCHOS includes the routine which chooses the highest scores from the score table to build a choice table, CHOT. Another routine reorders CHOT so that the highest index/score pair is the first entry. A more basic procedure uses the aforementioned routines to extract the highest score together with a second choice score if one exists. A flag is set if a close second choice is found.

The analysis provided in VALYZ consists mainly of the Breaux test or second look. Following a scheme devised by Dr. Breaux of NAVTRAEQUIPCEN, the two VRPs which are in contention for first choice recognition are compared to find those time slots which are significantly different. These rows are then correlated with the corresponding rows of the IFP. The technique effectively causes the pattern recognition algorithm to weigh the distinctive portions of the utterance more heavily than the similar portions. CHOT is reordered after this test and the confidence level is again investigated.

Special recognition processing is provided for phrase groups such as headings, wind, or missed approach phrases. Special masks are sent to RCGPP, RCGPT, and RCCRP for easy recognition of anticipated phrases. In the case of headings and wind, these masks single out the digits (and later "at" and wind speed for wind) for a first recognition pass. This boosts the probability of accumulating a set of phrases that concatenate to form a sensible message.

SPEECH UNDERSTANDING SUBSYSTEM. The Speech Understanding Subsystem has two major tasks: to send information to the speech recognition unit to help it decide on its choice of the correct phrase, and to take the phrases that the speech recognition sends and build messages out of them, if necessary. Refer to Figure 32 for the structure of this section.

The first part is performed by a task (ISAY) which, when trainee input is detected, sends a buffer full of information from the model controller to the speech recognition unit on CPU 2. This same information is sent to PMS (during phase 2) or the replay file (during a P-run) to assist in grading.

When the speech understanding unit receives a recognized phrase, it first builds and saves a record containing state-of-the-world information. This is done because other parts of the training system need information about the environment within which a message was issued. After this, a number of different message concatenators are called to look at the phrase and determine if it could be a part of a message that it is designed to build.

Concatenated Phrase Groups. SUS performs phrase concatenation for the following types of phrase groups. (The "..." indicates a required pause.)

a. Turn right heading / Turn left heading / Heading } ...X...X...X...

b. Wind ...X...X...X...at...X...

c. Missed approach ...C/S...map position...button X...

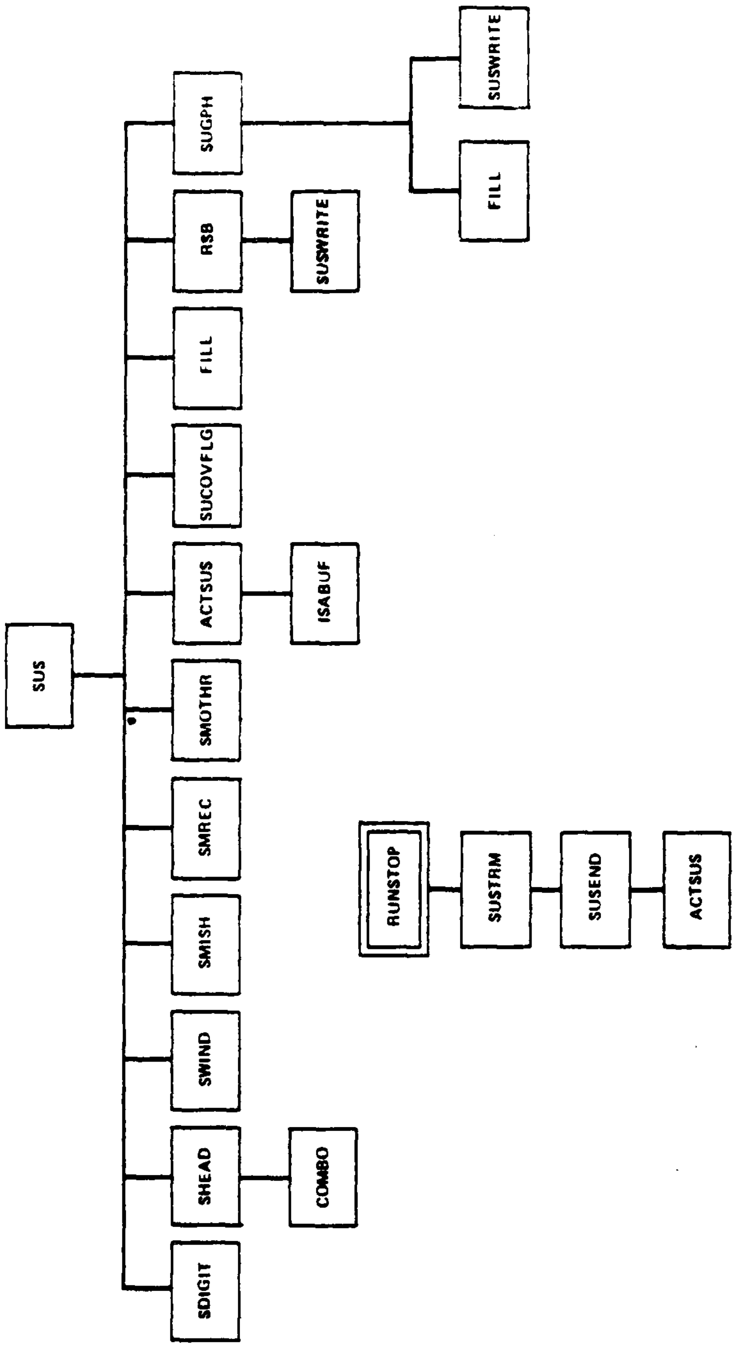


Figure 32. The Structure of the Speech Understanding Subsystem

d. Unrecognized phrase ...X...X...X...

e. X...X...X...

All other phrases are not concatenated.

A concatenated phrase group does not necessarily constitute a message. Phrase groups (a) and (c) are messages, whereas the remainder are not complete messages. Items in phrase group (d) can sometimes be considered as a message. Discussion of this and other probable confusions will be taken up shortly.

Full Phrase Groups. The key to SUS phrase grouping is the recognition of the first phrase of the group. For example, "Turn right heading," "Turn left heading," "Heading," "Wind," "Missed approach," any digit, or an unrecognizable phrase will trigger the phrase grouping mechanism. If the key is not detected, the phrases are not concatenated. For example, if "Wind" is not recognized or is misrecognized, the remainder of the phrases will not be treated as parts of a wind message.

If the key is present, concatenation is attempted. Examples of "perfect" phrase groups (i.e., those for which the trainee's R/T is correct and recognizable) follow:

a. Heading phrase group: "Army 876...Turn right heading...1...2...3..."

SSBF buffer is released:

Word 1: Bit settings vary

Word 2: LP4 time for digit "3" in half-second ticks

Word 3: 100ms offset from the above

Word 4: Phrase #: 105

Word 5: Heading: 123

Word 6: C/S: Bits 15-13 = 1
Bits 12-0 = all set to 1

Word 7: -1

Word 8: -1

Activity file record is constructed but not yet output:

Words 1 through 7 are the same as for SSBF buffer

Word 8: End of message bit reset

NOTES: If "over," then "correction" followed, the outputs would be:

(1) SSBF buffers:

(a) Released buffer containing "over"

(b) Released buffer containing "correction"

(2) Activity file record:

(a) The aforementioned activity file record is updated to reset the "over" and "correction" bits. The LP4 time is updated to the end of the phrase "correction." The record is then released.

(b) If any phrase other than "over" or "correction" had followed the last heading digit, the activity record would have been output without any of the changes stated in (2)(a).

b. Wind phrase group: "Wind...1...4...0...at...nine..."

SSBF buffer is released:

Word 1: Bit settings vary

Word 2: LP4 time for "nine" in ticks

Word 3: 100ms offset from the above

Word 4: Phrase #: 44

Word 5: Heading: 140

Word 6: Bits 15-13 = 7 (all set)
Bits 12-0 = 9

Word 7: -1

Word 8: -1

Activity file record is constructed but not yet output:

All words are the same as for SSBF buffer

The final output is as previously noted (dependent on following phrase(s)).

c. Missed approach group: "Missed approach...Marine 684...2 and 1/2 miles...button 1..."

SSBF buffer is released:

Word 1: Bit settings vary

Word 2: LP4 time for "button 1" in ticks

Word 3: 100ms offset from the above

Word 4: Phrase #: 25

Word 5: Missed approach position: 5*

Word 6: C/S: Bits 15-13 = 2
Button #: Bits 12-1 = 1

Word 7: -1

Word 8: -1

Activity file record is constructed but not yet output:

Words are the same as for SSBF buffer except for

Word 8: End of message bit reset

The final output form is as previously noted.

Concatenation group types (d) and (e) occur only as a result of some misunderstanding (or error). They possibly may be interpreted as being associated with a heading or turn heading, a wind heading, or as incorrect call sign digits (since SUS does recover a segmented call sign if the digits match those of the correct call sign).

d. Unrecognized phrase plus digits group:

"Air Force 307...????...1...1...0..."

SSBF buffer is released:

Word 1: Bit setting varies

Word 2: LP4 time for digit "0"

Word 3: 100ms offset from the above

*The missed approach mile entered is twice the actual to keep the contents of this word an integer.

Word 4: Phrase #: 108

Word 5: Digits: 110

Word 6: C/S: Bits 15-13 = 4
Bits 12-0 = all set to 1

Word 7: -1

Word 8: -1

Activity file record is constructed but not yet output:

Words are the same as for SSBF buffer

The final output form is as previously noted.

e. Digits group: "5...6...7..."

SSBF buffer is released

Buffer is as for unrecognized phrase plus digits group (d) except for:

Word 4: Phrase #: -1

Omissions and Unrecognized Phrases in Phrase Groups. If an omission is made after a concatenation has begun, the appropriate storage word remains undefined, i.e., all bits remain set (-1 if entire storage word). For example, if the trainee neglects to give the wind speed as part of the wind message, Bits 12-0 of Word 6 remain set to 1 in both buffers (SSBF and SSBFO). Improper syntax is not indicated in the activity record.

However, if an unrecognized phrase ("????") occurs when the wind speed is expected, Bits 12-0 are set to 0 in both buffers. This holds for all expected phrase group phrases with these exceptions:

a. If only one or two digits are recognized, a number is formulated using "O" as a placeholder for the unrecognized digit.

b. If none of the wind or heading digits are recognized, Word 5 of both buffers is set to -2 since the trainee conceivably may give a heading of "0...0...0...."

Probable Confusions. Whenever the trainee makes stylization errors, the potential for misinterpretation increases greatly. SUS accounts for four possible stylization errors:

a. (Turn right/left) heading X...X...X..., where the first phrase is recognized as a low confidence heading type phrase.

b. Army, Navy, Marine, Air Force ...X...X...X..., where the first utterance is not recognized and the digits correctly correspond to the actual call sign's digits.

c. X... (???? mile(s) from touchdown), where the digit corresponds to the correct mile and mile(s) from touchdown is low confidence or not recognized.

d. Wind...X...X...X..., etc., where the last two wind heading digits correspond to the correct wind heading and "Wind X" is recognized as "Wind" or low confidence "Wind."

Stylization errors may account for outputs of the nature "????...2... 3...," which possibly could start a wind message or a turn. SUS will not investigate the next phrase to determine the exact nature of the utterance.

In cases where the key phrase is not recognized, a message could end up being strung out. For example, if "wind" goes misrecognized and the rest of the wind phrase group is recognized, words of the ensuing buffers contain:

Next buffer 1: Wind heading digits in Word 5 of buffer, no additional phrase reference number information.

Next buffer 2: Phrase reference number for "at" in Word 4, no additional phrase reference number information.

Next buffer 3: Phrase reference number for the wind speed in Word 4, no additional phrase reference number information.

Digits associated with turn headings and wind headings will cause SUS to do the following:

Wind headings: Any 2 or 3 digits which match the wind heading suffice to cause insertion of the actual wind heading into word 5 of the buffers. The digits must be in the right order, but not necessarily the first choice digit recognition. Also, if one of the digits is recognized as being "too short" or not recognized at all and the other two digits are present, the correct heading is used.

Turn headings: A combination of digits which represent the turn which is closest to the model controller's turn is selected. Also the combination is selected to avoid a 360° turn. This selectivity holds only when second choice digits are recognized.

Turn selection: If both turns are presented as recognition choices, the correct turn (using the model controller as a standard) is given precedence unless the direction initiates a 360° turn.

Five mile rule: Outside of five miles, turn headings are chosen to be multiples of 5° if the combination is present.

Alternate Phrase Understood. Word 7 of SSBF and SSBFC is reserved for a second choice phrase. This is the phrase which is an alternate to the phrase identified in Word 4. SUS always fills Word 7 if the recognition module passes an alternate for the phrase in Word 4. However, the choices are screened by SUS, and SUS chooses the best phrase to be placed as the first choice.

If, after looking at the phrase, any concatenator indicates that it has a complete message, the phrase is checked for validity. If the message is accepted, it is put into an output buffer together with the state-of-the-world information saved earlier and is sent to APE, the model controller and PMS or the replay file.

AIRCRAFT/PILOT/ENVIRONMENTAL SIMULATION

One important objective of GCA-CTS is to provide the controller-trainee with a "realistic environment" in which to practice his new skills. GCA-CTS realizes this objective in the following ways:

a. It provides the trainee with a simulated PAR display.

b. A simulated "target" (the radar image of an aircraft) appears on the simulated PAR display.

c. The simulated target moves across the display in a manner which closely approximates the motion of the actual PAR image of a real aircraft on GCA.

(1) The simulated target's motion varies in response to controller-trainee advisories and other "approach events" in the same way as does the PAR image of a real aircraft on GCA whose pilot responds to advisories and other "approach events" by:

(a) Formulating a conception of the current "correct" rate-of-turn, rate-of-climb, and airspeed for the aircraft from the most recently received advisory or from the most recently encountered "approach event", in accordance with a body of specific rules and procedures which dictate "proper" behavior of pilots conducting GCA's.

(b) Attempting (with a degree of success dependent on the pilot's skill level) to manipulate the controls of the aircraft to achieve and maintain the above "correct" rate-of-turn, rate-of-climb, and airspeed.

(2) The simulated target's motion also varies in a manner similar to those motions of the PAR image of a real aircraft on GCA which appear attributable to the action of wind on the aircraft.

AD-A087 190 LOGICON INC SAN DIEGO CA TACTICAL AND TRAINING SYSTE--ETC F/G 17/9
GROUND CONTROLLED APPROACH CONTROLLER TRAINING SYSTEM (GCA-CTS)--ETC(U)
JUN 80 G BARBER, J BOLLENBACHER, D BREWTON N61339-77-C-0162
UNCLASSIFIED NAVTRAEQUIPC-77-C-0162-3 NL

AD
A087190

d. A simulated "pilot" emits utterances (using computer-generated speech) in response to controller advisories and other "approach events." The simulated pilot's "verbal" behaviors duplicate the verbal behavior of a real pilot conducting an actual GCA.

e. The simulated PAR display includes two numbers, labeled "wind speed" and "wind direction", which vary with time in a manner which closely approximates real wind speed and wind direction time histories, and which correspond to the wind speed and wind direction required to produce the wind-induced effects described in c(2) above.

Functions c, d, and e, above, are performed by that segment of the GCA-CTS software known as APE (Aircraft/Pilot/Environment).

In the broadest sense, APE acts within GCA-CTS as a "black box" to transform an input stream of simulated controller advisories (which it receives aperiodically from the GCA-CTS trainee-speech-recognition and trainee-speech-understanding routines) into an output stream of (1) aircraft position vectors and wind speed/wind-direction vectors (which it sends each 0.5 seconds as inputs to the GCA-CTS PAR-display-simulation routines), and (2) pilot "verbal" reply specifications, in the form of "VOTRAX" phrase-identification numbers (which it sends aperiodically as inputs to the GCA-CTS speech-generation routines). (See Figure 33.)

This transformation performed by APE involves four separate but concurrent processes which are embodied in the four APE software subroutines WIND, THINKPILOT, SPEAKPILOT, and MOVEPILOT. During an actual GCA-CTS approach simulation, a user-clock-driven executive subroutine, APEX, calls the above sequence of four subroutines once each 0.5 seconds.

WIND. This subroutine models a wind of predetermined intensity, direction, and variability, blowing across the approach track. Each time this subroutine is called it transmits a current windspeed/wind-direction vector to the GCA-CTS PAR-display-simulation routines, and it computes the current x- and z-axis components of the current wind for use in other APE routines.

THINKPILOT. The ongoing thought processes of the pilot are modeled in this subroutine. Each one of a number of such thought processes is embodied in a different individual subroutine called by THINKPILOT. Subroutine NEWADVISOR simulates the pilot detecting advisories embedded in the stream of incoming controller utterances. Subroutine PLTASSUMES simulates the pilot deciding (on the basis of a prolonged absence of incoming advisories) that he has lost radio contact with the controller and should execute a self-initiated waveoff. Subroutine PLTCOPIEDN simulates the pilot "comprehending" ("copying") the content of a newly-detected advisory and allows simulation of different pilot skill levels as manifested by varying frequencies of pilot failure to "copy" a detected advisory. Subroutine PLTDECIDES simulates the pilot monitoring the audibility of controller advisories and deciding whether he should verbally notify the controller that his transmissions are "weak but clear" as opposed to "loud and clear." Subroutine PLTWAVESHI simulates the pilot's decision, for any of a number of reasons, to execute a missed approach in spite of the

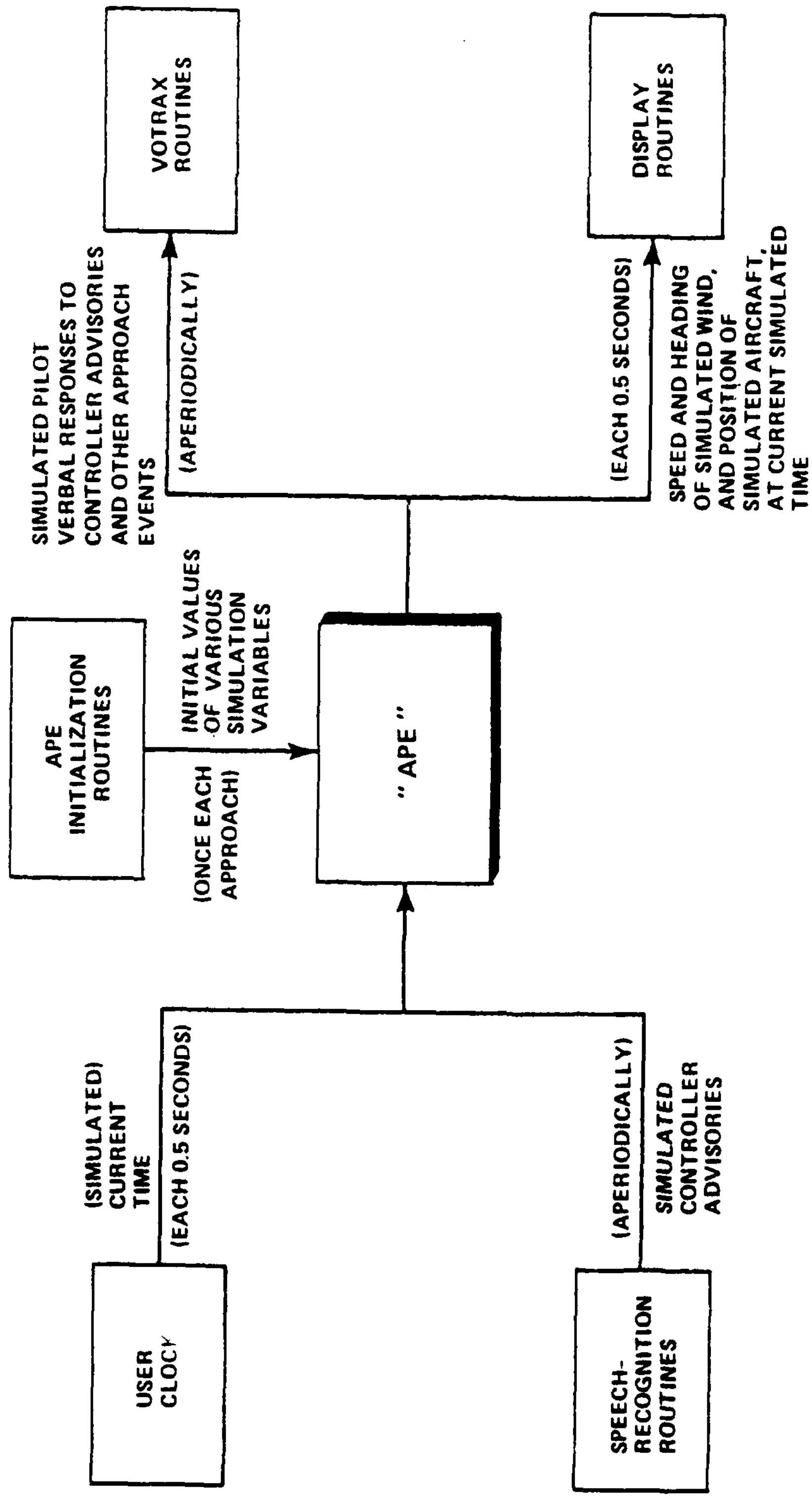


Figure 33. APE Role Within CCA-CTS

fact that the controller has not issued an advisory to that effect. Subroutine DEDUCETHEC simulates the pilot deciding what verbal reply, if any, to render in response to a newly-received advisory or a newly-encountered approach event of some other type, and deciding to delay rendering that reply until such time as it may be solicited by the controller uttering the word "over." And, finally, subroutine CONCEIVETH simulates the pilot reconceiving, each time the subroutine is called, a new current correct rate-of-turn, rate-of-climb, and airspeed for his aircraft, based on all information currently available to him (primarily, the most recently received advisory) and in accordance with all specified rules of pilot GCA behavior; and thereafter attempting to achieve and maintain the above correct rate-of-turn, rate-of-climb, and airspeed (i.e., attempting to fly his best guess of what currently constitutes a proper GCA). The subroutine allows simulation of different levels of pilot skill, manifested by different degrees of pilot accuracy in deducing a current correct rate-of-descent on the basis of all previously received glidepath advisories.

SPEAKPILOT. The speech process of the pilot is modeled in this subroutine. Each time this subroutine is called it transmits to the GCA-CTS speech-generation routines a request to generate whichever verbal reply the pilot (via THINKPILOT: DEDUCETH) currently thinks to be appropriate, if any.

MOVEPILOT. This subroutine models the pilot's actual (as opposed to attempted or correct) motor behavior and the dynamic response of the aircraft to the pilot's motor behavior. Each time the subroutine is called it transmits the current aircraft position vector to the GCA-CTS PAR-display routines. First, the current true value of rate-of-turn, rate-of-climb, and airspeed are computed by applying to the current correct values of those variables (as conceived by the pilot in THINKPILOT: CONCEIVETH) certain error-inducing processes which embody, in a single, integrated (and indecomposable) model, both (1) the pilot's skill level in achieving and/or maintaining (via his motor behavior) any specific instrument picture he may desire, and (2) the sensitivity of the dynamic response of the aircraft type being simulated to the pilot's motor behavior. (The above two phenomena are not modeled independently in APE.) Next, the aircraft's actual (true) rate-of-turn is integrated with respect to the time required to determine the current aircraft heading with respect to the frame of reference of the simulation coordinate axes. Wind velocity, aircraft heading, and true airspeed are then used to determine the aircraft's current velocity with respect to the surrounding air mass. That velocity is then resolved into x-, y-, and z-axis components, to which are added, correspondingly, the x- and z-axis components of the current wind velocity, yielding the aircraft's current velocity with respect to the simulation coordinate system frame of reference. This velocity vector is integrated with respect to time over a 0.5 second period to generate a displacement vector which, when added to the last-computed aircraft position vector, yields the current aircraft position vector.

APE's subroutine call structure is depicted in Figure 34, and the functional interactions and general flow of information between the various routines described above are summarized in Figure 35. Appendix I provides a glossary of APE local variables.

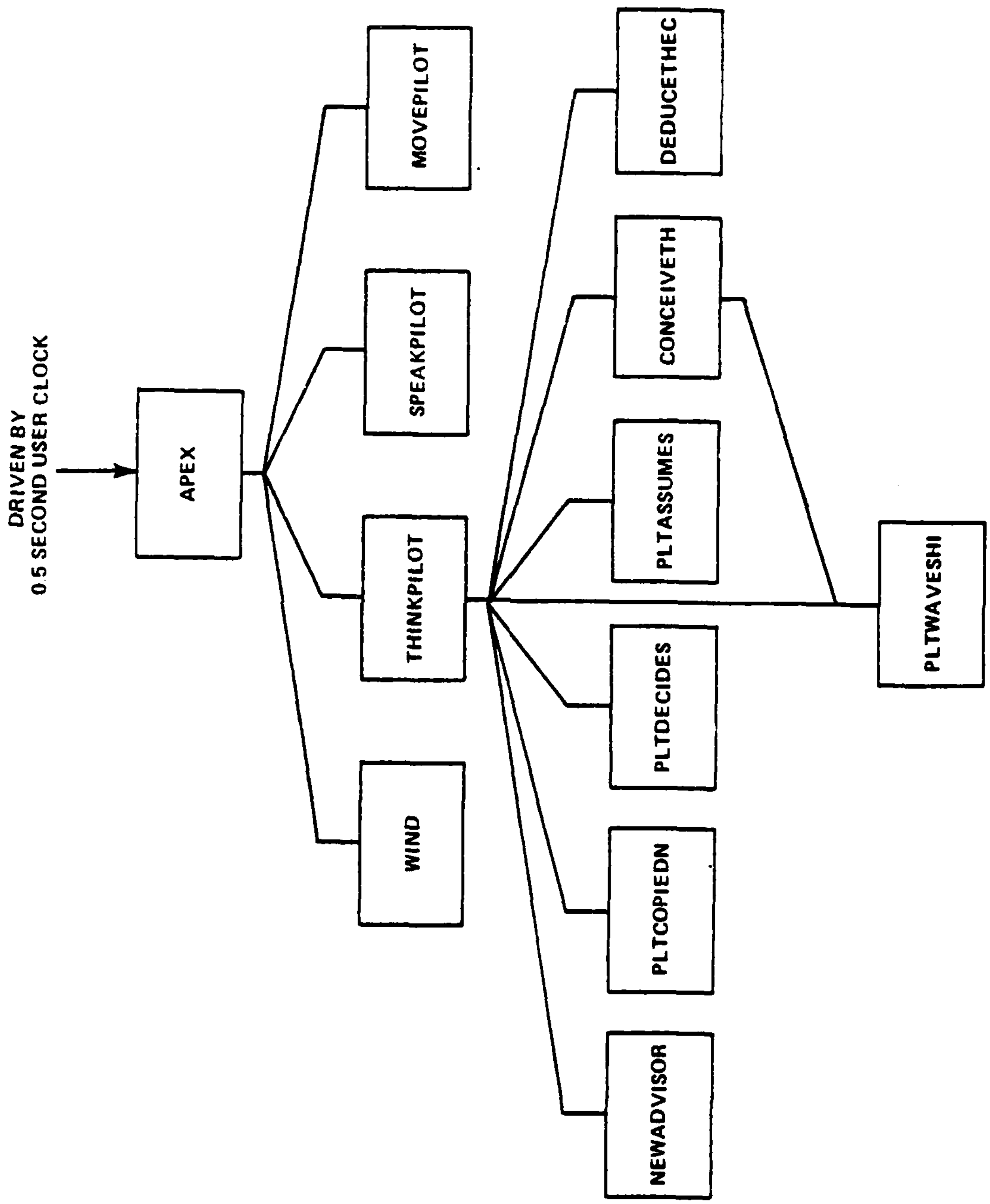


Figure 34. APE Subroutine Call Structure

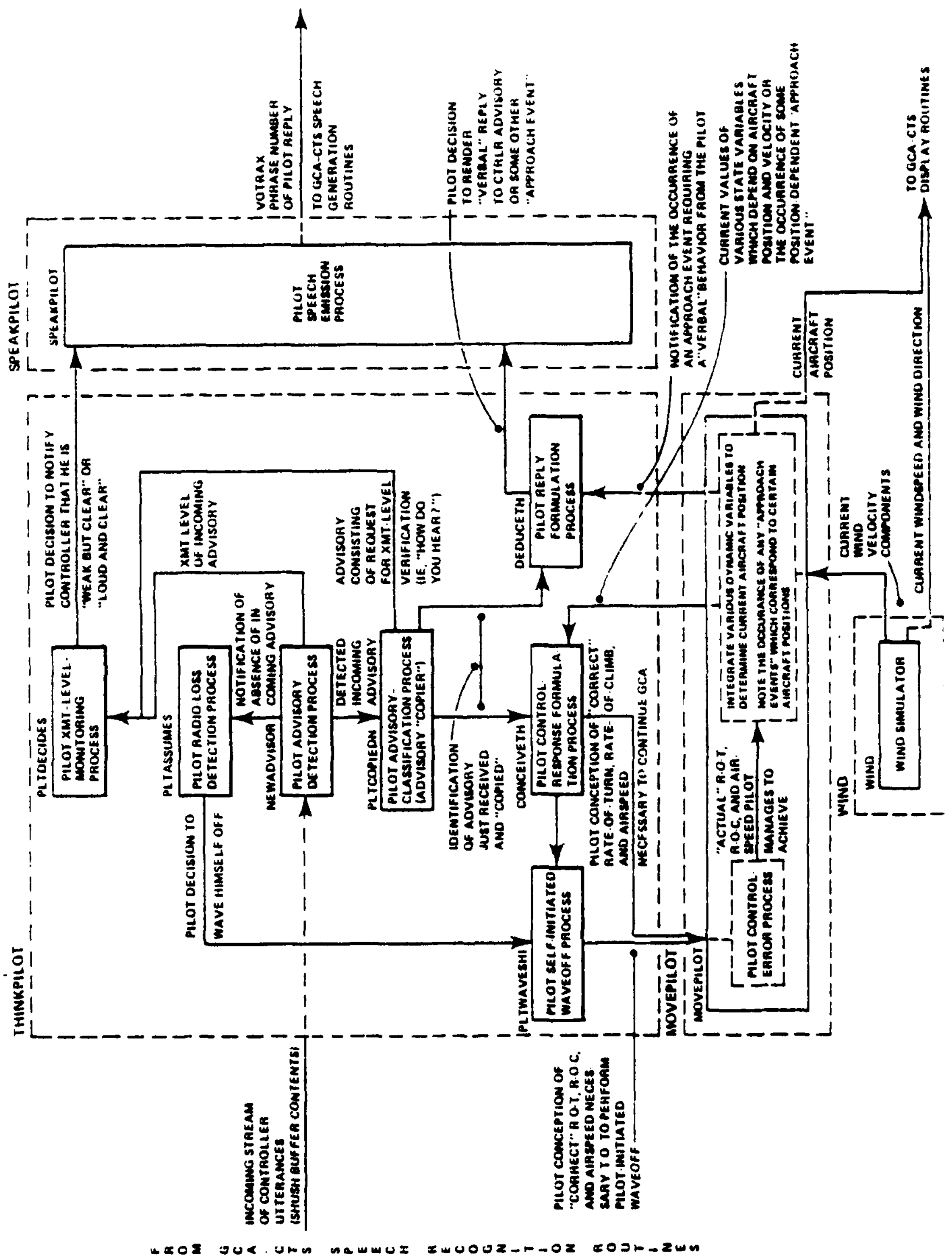


Figure 35. Information Flow Within APE

THE WIND MODEL. Some background is required before this model is discussed. First, there is a resident random number generator accepting a seed value and providing a random integer between $-(2^{15}-1)$ and $+(2^{15}-1)$, inclusive, uniformly distributed over that interval and uncorrelated from sample to sample. In the following these random numbers are called RN, and the implication is that each time this symbol occurs a new random number is to be found and its value used. For example, the formula:

$$r = (RN+RN+RN)/32767$$

means that r is formed by summing three successive samples of the random number generator and dividing by 32767.

Secondly, the units of length, time, and angle internal to the simulation are feet, seconds, and radians.

Finally, the coordinate system adopted has its origin at the intersection of the glideslope with the ground, and its z axis horizontal and opposite to the direction of approach. The x and y axes are horizontal to the pilot's right, and vertical, respectively. A perfect approach is in the y-z plane, moving toward the origin, as illustrated in Figure 36.

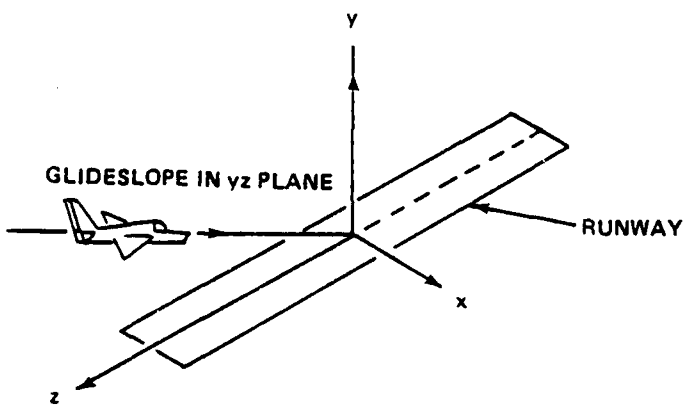


Figure 36. Coordinate System

Wind is modeled as the sum of a steady component and a random component, modified by gusts. The random component is modeled as the combination of two uncorrelated processes acting along and across the steady wind vector. Each of these processes has an auto covariance as a function of time. The result of this auto correlation is that successive samples of wind (taken each half second) are not independent, thereby preventing wild variations in wind velocity and direction.

Gustiness is modeled as occasional increases and decreases in the wind, affecting the component along and across the steady wind direction equally. Gusts and "antigusts" (decreases in wind intensity) are assumed to occur equally frequently, with equal average durations. The steady wind speed is assumed to be equal to the geometric mean of the mean wind speed during gusts and antigusts along the steady wind direction. (So gusts of +100 percent intensity are accompanied by antigusts of -50 percent.)

During gusts, antigusts, and no gusts, the wind variability is assumed to be twice as great along the steady wind component as across it.

The wind model is determined by the external parameters listed in Table 13.

The model equations follow. They are executed once each half second.

1. $r = (RN+RN+RN)/32767$
2. $W_{1\ new} = \alpha W_{1\ old} + \beta r$
3. $r = (RN+RN+RN)/32767$
4. $W_{2\ new} = \alpha W_{2\ old} + \beta r$
5. $R = RN$

Note: $W_{1\ old} = 0$ and $W_{2\ old} = 0$ on the first pass through the algorithm.

6. If state is not NOGUST then:

   If $R < N_3$ then: $\left\{\begin{array}{l}\text{State = NOGUST}\\ \text{go to 9}\end{array}\right\}$

7. If $R < N_1$ then: $\left\{\begin{array}{l}\text{State = GUST}\\ \text{go to 9}\end{array}\right\}$

8. If $R < N_2$ then: $\left\{\begin{array}{l}\text{State = ANTIGUST}\\ \text{go to 9}\end{array}\right\}$

9. $W_z = S_W\ (\text{State})\ (k_1 + 2k_3 W_1 - k_4 W_2)$

   $W_x = S_W\ (\text{State})\ (k_2 + 2k_4 W_1 + k_3 W_2)$

The outputs of the wind model are W_z and W_x, the instantaneous components of wind in the z and x directions, and current windspeed (ENWSP, in knots) and wind heading (ENWHDG, in degrees), both truncated to the nearest lesser integer, which are computed using the values of W_x and W_z determined above. Steps 1 and 3 of this model form a pseudo-Gaussian random variable with zero mean and unit variance. It is limited to the interval [-3, +3] and has the distribution illustrated in Figure 37. Steps 2 and 4 cause sample-to-sample correlation of the (otherwise uncorrelated) pseudo-Gaussian variables r. Steps 6, 7, and 8 implement a three-state Markov process which has states labeled NOGUST, GUST and ANTIGUST. The latter two states each

TABLE 13. WIND MODEL PARAMETERS

Sub-Routine	Parameter Designation	Definition
ENRH	H_R	Runway heading. (True azimuth of the direction in which a plane lands.) (Input as degrees and immediately converted to radians.)
ENWHT	H_W	Mean heading of the wind. (True azimuth of the direction from which the wind comes.) (Input as degrees and immediately converted to radians.)
ENMWS	S_{WN}	Nominal windspeed. (Mean speed of the wind in the absence of gusts and along its mean heading.) (Input as kts; converted to ft/sec immediately upon input [see "S_N" below].)
ENWVP	V	Wind variability parameter. If V=0 the wind is steady. If V=1 the standard deviation of the wind in its mean direction is 1/3 S_{WN}. The variability of the orthogonal direction is 1/2 as great; i.e., with V=1 the standard deviation is 1/6 S_{WN}. (dimensionless)
NWSCT	t_w	Windspeed correlation time. (The time lag at which the auto-covariance of wind speed is 1/2 times its variance.) (seconds)
NMGS	S_{WG}	Wind gust speed. (The mean wind speed in its mean direction during gusts.) (Input as kts; converted immediately upon input to ft/sec [see "S_G" below].)
NGOCC	F_G	Fraction of the time gusts occur. (Gusts and antigusts occur equally often and for equally long periods of time, therefore, $0 \le F_G < 1/2$.) (dimensionless)
NMGD	T_G	Mean duration of a wind gust or antigust (seconds). Must be larger than 0.5 second.

The runtime parameters used in the wind model are as follows:

ENWHR	W	$= (H_R - H_W) \cdot \mathrm{mod}\ 2\pi$. Wind direction relative to z axis.
ENCOS	k_1	$= \cos \omega$
ENSIN	k_2	$= \sin \omega$
ENALPHA	α	$= e^{-\frac{1}{2}t_w}$
ENBETA	β	$= 1-\alpha$

TABLE 13. WIND MODEL PARAMETERS (CONT)

Sub-Routine	Parameter Designation	Definition	
ENK3	k_3	$= \sqrt{\frac{1+\alpha}{1-\alpha}}\ \frac{v}{6} \cos \omega$	EN2K3 = $2k_3$
ENK4	k_4	$= \sqrt{\frac{1+\alpha}{1-\alpha}}\ \frac{v}{6} \sin \omega$	EN2K4 = $2k_4$
ENMWS	S_N	$= 1.6887\ S_{WN}$	Equivalent to ENSW(1) ENSW(2) ENSW(3)
ENMGS	S_G	$= 1.6887\ S_{WG}$	
ENMAGS	S_{AG}	$= S_N^{\ 2}/S_G$	
ENN1	N_1	$= 32767\ \left(\frac{F_G}{T_G(1-2F_G)} - 1\right)$	
ENN2	N_2	$= 32767\ \left(\frac{2F_G}{T_G(1-2F_G)} - 1\right)$	
ENN3	N_3	$= 32767\ \left(\frac{1}{T_G} - 1\right)$	
ENWX ~	W_x	R ~ r	ENW1 ~ W_1'
ENWZ ~	W_z	ENSW ~ S_w	ENW2 ~ W_2

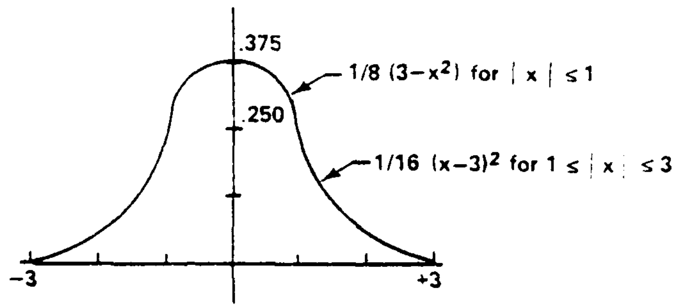


Figure 37. Probability Density of the Pseudo-Gaussian Random Variable Used to Form Wind Components

occur fraction F_G of the time, with average duration T_G seconds at each occurrence. Step 9 produced the headwind and crosswind components of the wind.

The significance of some of the external parameters of the wind simulation is best understood through example. The simulation has therefore been exercised with two different sets of parameters and the results plotted to illustrate the influence of the less obvious parameters.

Persons unfamiliar with the statistical treatment of sequences of numbers may find correlation time a new concept. Correlation time is a measure of how quickly a randomly varying value may change. If the wind speed is 5 knots greater than its average value at one instant, it will be almost 5 knots greater than its average value one microsecond later, showing that wind speed has a correlation time larger than one microsecond. In general, samples from a random process with correlation time t are very similar when the samples are taken at times differing by a small fraction of the correlation time, and they are essentially independent when the time interval between samples is large compared to the correlation time. The effect is illustrated in Figure 38. Thirty-one samples of two random processes are plotted in that figure. Each process has a mean value of zero and an approximately Gaussian distribution of values. (The standard deviation of each process is indicated in the figure.) The first process is a sequence of entirely independent samples, so that successive values are uncorrelated and the correlation time is zero. The second process (which was derived from the first by passing it through a first order digital filter) has a correlation time of six samples. Notice that when the second process takes on a low value, it tends to remain low, and the sample-to-sample variation tends to be a smaller fraction of the standard deviation of the process than for an uncorrelated sequence.

The most pronounced effect of increasing correlation time is to reduce the frequency with which the process crosses its mean value. In thirty-three samples of an uncorrelated process, the average number of crossings will be sixteen, as indeed occurred in the sample of Figure 38. The two samples of simulated wind plotted in Figures 39 and 40, have correlation times of 8 and

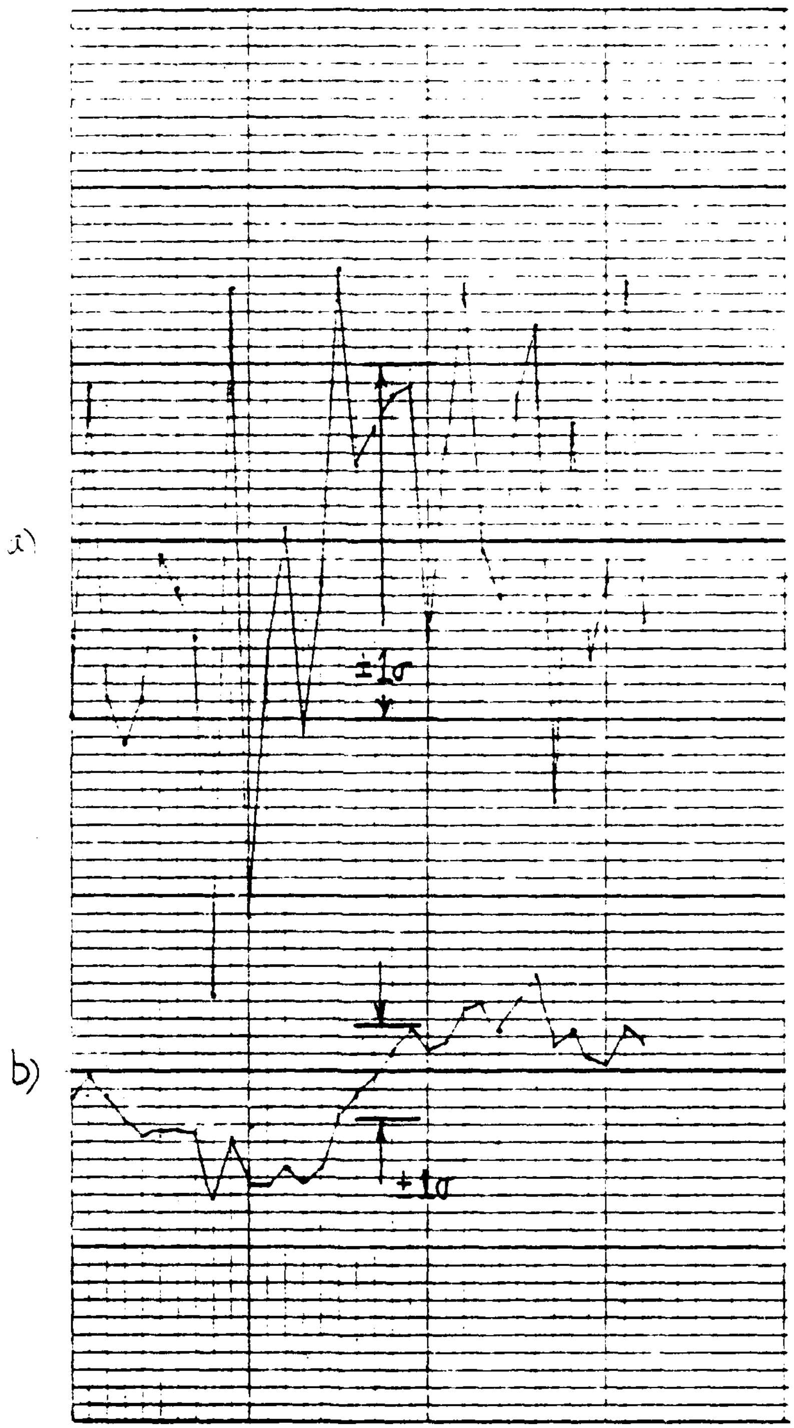


Figure 38. Random Sequences with Different Correlation Times
a) Correlation time t = 0.
b) Correlation time t = 6 sample times

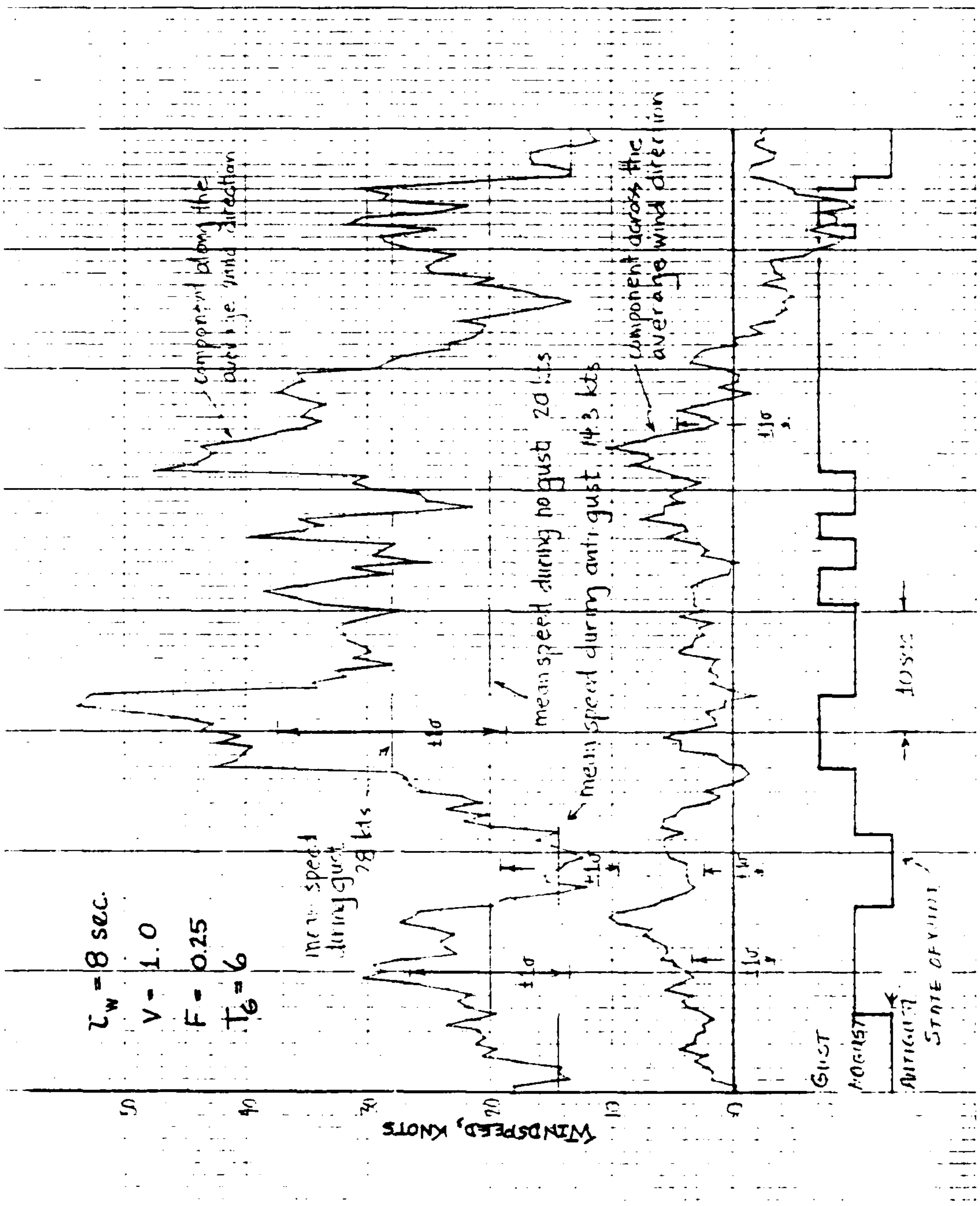


Figure 39. Sample of Simulated Wind with Correlation Time of 8 Seconds

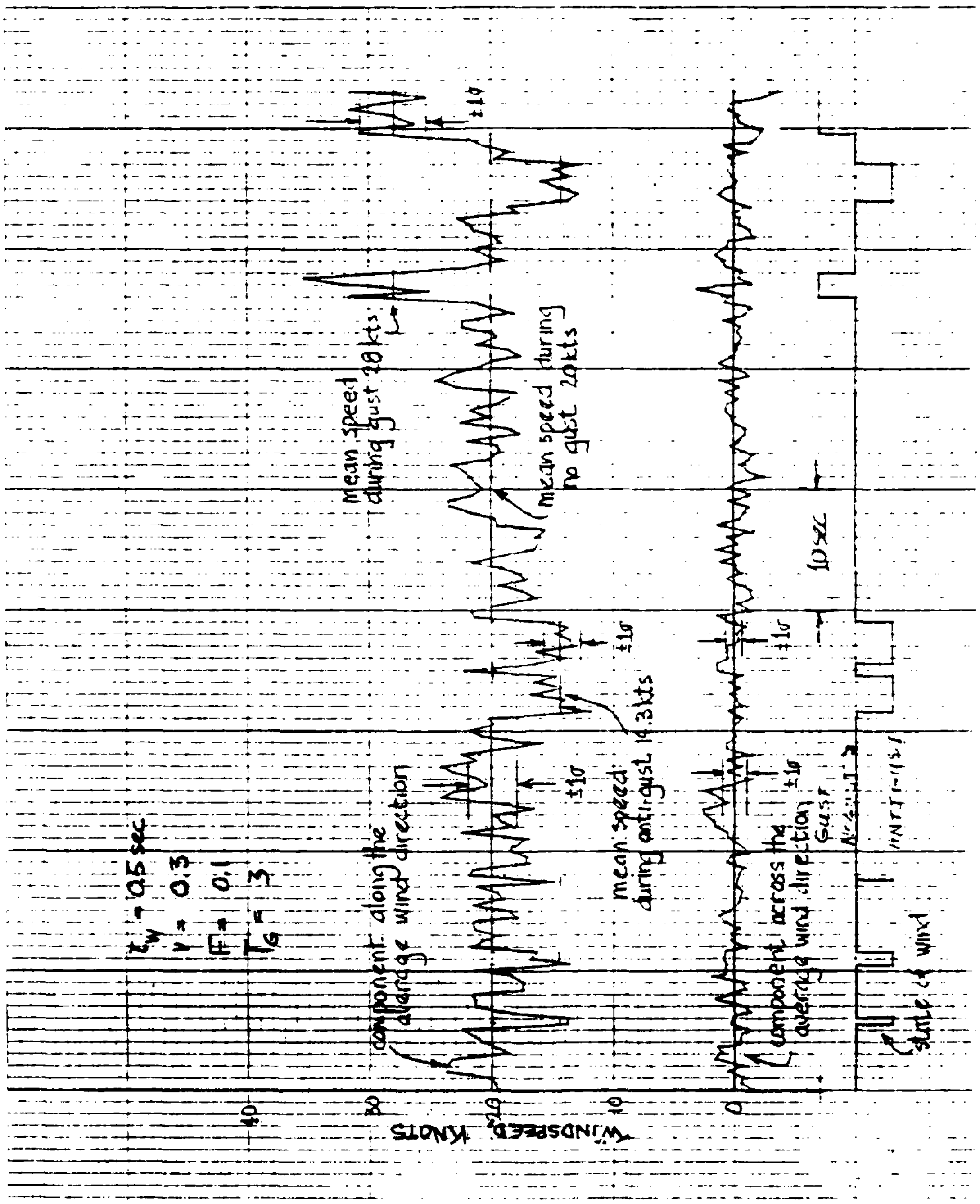


Figure 40. Sample of Simulated Wind with Correlation Time of .5 Seconds

0.5 seconds, respectively. The components both along and across the average wind direction are plotted in these figures. The mean value of the cross component is zero, and it is quite clear that the longer correlation time results in fewer zero crossings for that component. The same effect can be observed in the other component, although less clearly because the mean value differs in the gust, no-gust, and antigust states.

The frequency of mean-value crossing is controlled entirely by the process correlation time and is not dependent on the process standard deviation. The large number of zero crossings of the cross component in Figure 40 is thus properly attributed to the short correlation time (0.5 seconds) and not to the small standard deviation of that process. The standard deviation, on the other hand, is controlled by the variability parameter, V, and is entirely independent of the correlation time.

Another effect of correlation time on the wind simulation is to introduce sample-to-sample correlation of wind direction, as illustrated in Figure 41. The tip of the two-dimensional wind vector has been plotted in that figure,

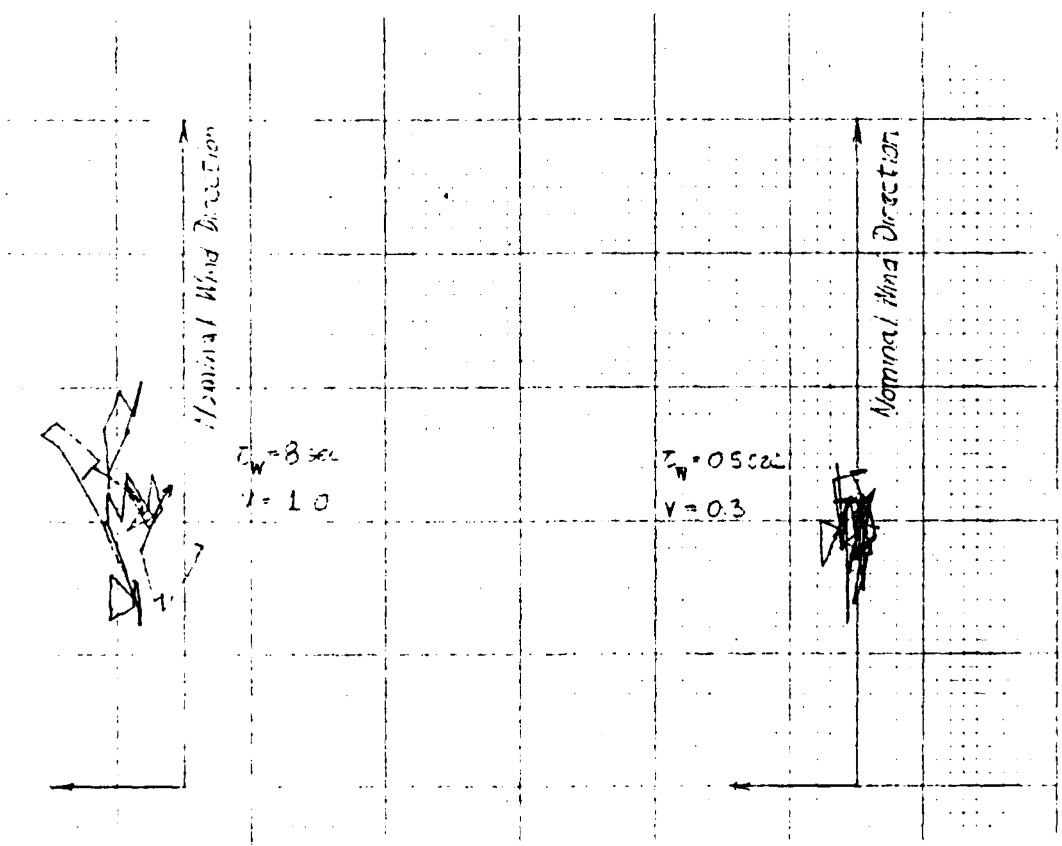


Figure 41. Two-Dimensional Plots of Simulated Wind Samples

showing the variation of the wind for a period of about twenty seconds. The data were taken from the same simulation used to generate Figures 39 and 40. Notice that during this interval the larger correlation time results in the wind dwelling to the left of its average direction.

The influence of the variability parameter, V, is also apparent in these examples, since the standard deviation of the wind's variability about its mean value is 30 percent as great in the second example as in the first, resulting from V values of 1.0 and 0.3.

The state of the wind simulation (gust, no-gust, or antigust) is also plotted in Figures 25 and 26 to illustrate the influence of the gust parameters. In the first example (Figure 38), the parameter F_G is given a value of 0.25, indicating that, on the average, the wind should gust 25 percent of the time and antigust 25 percent of the time, leaving 50 percent of the time for "normal" windiness. The corresponding parameter value in the second example is 0.1, leading to a non-gusting condition 80 percent of the time. Notice that the wind states in these two examples are consistent with these average percentages, and yet neither the duration of any state nor the alternation between states is predictable.

Comparison of Figures 38 and 39 also illustrates the influence of the mean gust duration parameter, T_G. In the first case the parameter value is 6 seconds and gusts and antigusts lasted for 2 to 19 seconds, whereas in the second example the parameter value is 3 seconds and the gusts and antigusts lasted for from 0.5 to 3.5 seconds. In general, the Markov process used to model the gustiness will lead to an exponential distribution of gust and antigust durations, and the standard deviation of the duration of these states will therefore be the same as their mean value: an indication that wide variability in gust duration is a characteristic of the model.

Further examination of these examples verifies other features of the wind simulation, for example,

a. The variability of the "along" component of wind (as indicated by its standard deviation in each state) is proportional to the mean value in that state and the variability parameter, V.

b. When the variability parameter, V, is unity, the standard deviation of the "along" component of the wind is one-third its no-gust average value. (A negative "along" component would therefore be a three sigma event when V=1.)

c. The variability of the "across" component of the wind is always one-half that of the other component.

d. When the difference between mean gust speed and mean no-gust speed (S_{WG}-S_{NG}) is small compared to the wind's variability in the no-gust state (σ=VS_{NG}), it is difficult to deduce the presence of gusts or antigusts from the windspeed history. The presence of these states is further obscured if the mean gust duration is similar to, or shorter than, the wind correlation time.

THE PILOT THOUGHT PROCESS MODEL. At any given time during a real GCA approach, the pilot's "thought process" may be characterized as an attempt to deduce (on the basis of the time-history of "approach events" he has thus far observed) the current "correct" rate-of-climb, rate-of-turn, and airspeed for his aircraft, i.e., that rate-of-climb, rate-of-turn, and airspeed which conforms to those understood "rules" for conducting GCA approaches which apply to the pilot's assumed current circumstances. Furthermore, it appears that, by defining the set of "approach events" carefully, we may view the pilot's "thought process" as a fairly straightforward, deterministic process, in which the pilot's concept of the "correct" rate-of-climb, rate-of-turn, and airspeed remains constant "between" successively encountered "approach events," but changes instantaneously whenever a new "approach event" is encountered. This change depends only on the type of "approach event" encountered and the current values of a small number of "stated variables" representing conditions like "aircraft currently climbing out on missed approach," etc.

APE simulates the pilot's "thought process" by representing as real-valued variables the pilot's current concept of the "correct" rate-of-climb (PTYDM), "correct" rate-of-turn (PTHDM), and "correct" airspeed (PTASM), and by assigning new values to those variables whenever APE detects the occurrence of a new "approach event." Due to the "deterministic" nature of the GCA "rules" structure that APE assumes, the portion of APE which implements this process consists of a set of algorithms standing essentially in a one-to-one correspondence to the set of distinct "approach event" types. Upon encountering a new "event" of a given type, APE merely branches to the corresponding algorithm, where the new values of PTYDM, PTHDM, and PTASM are computed. The algorithms also make use as "input" of the current values of the stated variables PTCLO, PTTR, PTTL, PTTRH, PTTRWM, PTTLWM, PTNGR, PTDES, PTWMAH, PTHDASS, PTYDI, and PTGPADSBL, where real GCA analogues (if any) are noted in the listing comments of common block PLT.CO.

The detection of, and pilot response to, a small number of types of "approach events" related to aircraft position and/or altitude (e.g., "aircraft has just reached assigned altitude") are simulated in the routine MOVEPILOT. The majority of "approach event" types are handled in the routine THINKPILOT; in particular, the detection of, and pilot "mental" response to, incoming GCA advisories (which are, of course, "approach events") are simulated there.

In the case of "incoming advisory"-type "approach events" only, it should be noted that not every such "event" enters into the pilot's simulated thought process. Pilots conducting real GCAs are known to occasionally fail to copy an incoming advisory. APE simulates this phenomenon in subroutine PLTCOPIEDN of THINKPILOT by deleting from the stream of advisories (spoken by the GCA-controller-trainee, recognized by SUS, and transmitted to APE) a fraction of all glidepath/course position/trend advisories, which fraction varies with pilot skill level (PTYP) according to Table 14.

TABLE 14. PERCENTAGE OF ADVISORIES COPIED AS A FUNCTION OF PILOT SKILL LEVEL

Pilot Skill Level (PTYP)	Percentage of Advisories Copied*
1	98
2	90
3	80
4	70
5	60

*PTNOCOPY = FNOCOPY (PTYP)

The following subsections describe the changes in the pilot's concept of the current "correct" rate-of-climb, rate-of-turn, and airspeed upon encountering each of the different "approach events." The corresponding algorithms lie in APENIT (1), THINKPILOT (2-17) and MOVEPILOT (18-20).

Approach Simulation Commences. The pilot's concept of "correct" rate-of-climb, rate-of-turn, and airspeed changes as follows:

Rate-of-Climb (PTYDM, feet/second (converted to feet/minute))

Rate-of-Turn (PTHDM, radians/second (converted to degrees/second))

Airspeed (PTASM, feet/second)

"Correct" value of airspeed depends on the aircraft type being simulated, as listed in Table 15.

TABLE 15. AIRCRAFT APPROACH AIRSPEEDS

Aircraft Type (ACTYP)	Final Approach Airspeed (knots)*
1	98
2	115
3	130
4	156

* = SFAAS (ACTYP), knots = PTASFA, feet/second

<u>Pilot Copies "Begin Descent" or Copies a Glidepath Position or Trend Advisory Before Copying "Begin Descent".</u> The pilot now conceives the current "correct" rate-of-climb, rate-of-turn, and airspeed as follows:

<u>Airspeed</u> (PTASM)

Unaffected

<u>Rate-of-Turn</u> (PTHDM)

Unaffected

<u>Rate-of-Climb (PTYDM, feet/second)</u>

Pilot's concept of current "correct" rate-of-climb is dependent on aircraft type as shown in Table 16. These are ideal, no-wind rates of descent for aircraft on a 3° glideslope having final approach airspeeds shown in Table 16.

TABLE 16. STANDARD RATES OF CLIMB

Aircraft Type (ACTYP)	Final Approach Rates of Climb (feet/minute)*
1	-520
2	-608
3	-688
4	-807

* = SYDI (ACTYP), = PTSYDI, feet/second

<u>Pilot Copies Glidepath Position or Trend Advisory After Copying "Begin Descent."</u> The pilot conceives the current "correct" rate-of-climb, rate-of-turn, and airspeed as follows:

a. Pilot formulates an estimate, $\Delta y(t)$, of his current vertical displacement from the glidepath, based on his assumed range from touchdown and the current advisory type; the variance, $V_{\Delta y}(t)$, associated with this estimate is a function $v_{\Delta y}(t) = \alpha(\Delta y(t), z(t))$ of $\Delta y(t)$ and the current range from touchdown $z(t)^2$.

---

[1] Namely, $\frac{1}{12}\left[\overbrace{\text{PTDLYVAR(GPZONE)}}^{\text{zone width as part of blipheight in real space}} * \overbrace{.02269 * (\text{ACZ}+3605.07)}^{\text{blipheight in real space}}\right]^2$

b. Pilot next formulates an estimate $\dot{y}_E(t)$ of his "recent" "true" rate-of-climb; if last prior glidepath advisory was copied at time $t-\Delta t$, then:

$$\dot{y}_E(t) = \frac{\Delta y(t) - \Delta y(t-\Delta t)}{\Delta t}$$

and associated variance is

$$v_{\dot{y}_E}(t) = \frac{v_{\Delta y}(t) + v_{\Delta y}(t-\Delta t)}{\Delta t^2}$$

c. The pilot's current indicated rate-of-climb, $\dot{y}_i(t)$, and his current estimated "true" rate-of-climb $\dot{y}_E(t)$, together constitute a current estimate of the "descent-parallel-to-glidepath" rate-of-climb $\dot{y}_{P,E}(t)$, according to:

$$\dot{y}_{P,E}(t) = \dot{y}_i(5) - \dot{y}_E(t)$$

with associated variance

$$v_{\dot{y}_{P,E}}(t) = v_{\dot{y}_E}(t)$$

d. If the pilot then asumes the "true" "descent-parallel-to-glidepath rate-of-climb" $\dot{y}_P(t)$ may be expressed as a linear combination of the current estimate $\dot{y}_{P,E}(t)$ and the most recent prior value of $\dot{y}_P$, $\dot{y}_P(t-\Delta t)$,

$$\dot{y}_P(t) = \alpha\dot{y}_P(t-\Delta t) + (1-\alpha)\dot{y}_{P,E}(t)$$

Then choosing α to minimize the variance $\sqrt{\dot{y}_P(t)}$ associated with $\dot{y}_P(t)$, we have:

$$\alpha = \frac{v_{\dot{y}_{P,E}}(t)}{v_{\dot{y}_{P,E}}(t) + v_{\dot{y}_P}(t-\Delta t)}$$

e. Given this value of α, we may immediately deduce the pilot's current concept $\dot{y}_P(t)$ of the "descent-parallel-to-glidepath rate-of-climb"; i.e., that rate-of-climb which, were the pilot to instantaneously achieve it and thereafter maintain it, would cause his aircraft to descend on a 3° glide-slope. Descent would be parallel to the glidepath if the pilot concurrently maintained the correct track in the azimuth plane).

f. Likewise, we may compute $v_{\dot{y}_p}(t)$ according to:

$$v_{\dot{y}_p}(t) = \alpha^2 v^2_{\dot{y}_p}(t-\Delta t) + (1-\alpha)^2 v^2_{\dot{y}_{p,E}}(t)$$

g. The pilot now conceives the current "correct" rate-of-climb, rate-of-turn, and airspeed as follows:

<u>Airspeed</u> (PTASM)

Unaffected

<u>Rate-of-Turn</u> (PTHDM)

Unaffected

<u>Rate-of-Climb</u> (PTYDM feet/second)

Pilot's concept of current "correct" rate-of-climb changes as shown in Table 17.

TABLE 17. RATE-OF-CLIMB CHANGES WITH GLIDEPATH ADVISORIES

Advisory	Current "Correct" Rate-of-Climb (feet/minute) (=PTYDM, feet/second)
well below	$\dot{y}_p(t)$ + 500
below	$\dot{y}_p(t)$ " + 300
slightly below	$\dot{y}_p(t)$ " + 100
on glidepath	$\dot{y}_p(t)$ " 0
slightly above	$\dot{y}_p(t)$ " - 100
above	$\dot{y}_p(t)$ " - 300
well above	$\dot{y}_p(t)$ " - 500
going above	$\dot{y}_p(t)$ " - 100
going further above	$\dot{y}_p(t)$ " - 300
going below	$\dot{y}_p(t)$ " + 100
going further below	$\dot{y}_p(t)$ " + 300

The correspondences between symbols in the preceeding discussion and identifiers in the APE Fortran implementation are shown in Table 18.

TABLE 18. CORRESPONDENCE BETWEEN THINKPILOT SYMBOLS AND FORTRAN VARIABLES

THINKPILOT Symbols	FORTRAN Variables
$\Delta y(t)$	EDELY
$v_{\Delta y}(t)$	EVARDELY
$\Delta y(t-\Delta t)$	PTOEDELY
$v_{\Delta y}(t-\Delta t)$	PTOEVARDELY
$\dot{y}_E(t)$	EYDI (early reference)
$v_{\dot{y}_E}(t)$	EVARYDI
$\dot{y}_i(t)$	ACYD
$\dot{y}_{P,E}(t)$	EYDI (last reference)
α	ALPHA
$(1-\alpha)$	GAMMA
$\dot{y}_P(t-\Delta t)$	PTYDI (early reference)
$\dot{y}_P(t)$	PTYDI (last reference)
$v_{\dot{y}_{P,E}}(t)$	PTVARYDI (early reference)
$v_{\dot{y}_P}(t-\Delta t)$	PTVARYDI (last reference)

Because APE may process as many as two SUS buffers in each real-time 0.5 seconds, it is possible – though not likely – that two successive glidepath advisories might occur "simultaneously" from the point of view of the THINKPILOT clock, PTEYCLK. In such cases, APE treats Δt in the above equation as = 0.5 seconds.

Pilot Copies "Turn Left/Right Heading XXX". If a gyro-failure is in progress, (stated variable ACGYRO = .false.; (i.e., pilot has copied "this will be a no-gyro approach" at some earlier time during the approach) the pilot disregards this approach event. Under these circumstances the pilot's concept of the current "correct" rate-of-climb, rate-of-turn, and airspeed changes as follows:

Airspeed (PTASM)

Unaffected

Rate-of-Climb (PTYDM)

Unaffected

Rate-of-Turn (PTHDM, radians/second)

If a climbout is not in progress (PTCLO = .false.), "correct" rate-of-turn changes to ±3.0 degrees/second if assigned rate-of-turn (PTHDASS, radians/second) = 3.0 degrees/second, or to ±1.5 degrees/second if assigned rate-of-turn = 1.5 degrees/ second, until reaching heading XXX degrees, at which time "correct" rate-of-turn will change to 0 degrees/second (i.e., PTHMTN ← XXX).

If a climbout is in progress for any reason (PTCLO = .true.) the "correct" rate-of-turn changes immediately to 0.0 degrees/ second, but upon reaching the assigned altitude (PTYMTN) at termination of this climb, "correct" rate-of-turn will change to ±3.0 degrees/second until reaching heading XXX degrees, at which time "correct" rate-of-turn will change to 0.0 degrees/second (i.e., PTWMAH ← XXX).

Pilot Copies "Make Half-Standard-Rate Turns". "Assigned rate of turn" stated variable PTHDASS is changed to .0261798 radians/second (1.5 degrees/second). The pilot's concept of "correct" current rate-of-climb, rate-of-turn, and airspeed changes as follows:

Airspeed (PTASM)

Unaffected

Rate-of-Climb (PTYDM)

Unaffected

Rate-of-Turn (PTHDM, radians/second)

Unaffected if currently zero; otherwise, halved. (The "correct" turn rate concept is altered immediately even if currently executing a turn.)

Pilot Copies "Turn Left/Right". If a climbout is in progress for any reason (PTCLO = .true.), the pilot disregards this approach event. In other cases, the pilot's concept of the current "correct" rate-of-climb, rate-of-turn, and airspeed changes as follows:

Airspeed (PTASM)

Unaffected

Rate-of-Climb (PTYDM)

Unaffected

Rate-of-Turn (PTHDM, radians/second)

"Correct" rate-of-turn changes to ±3.0 degrees/second if assigned rate of turn (PTHDASS, radians/second) = 3.0 degrees/second; or to ±1.5 degrees/second if assigned rate of turn = 1.5 degrees/second.

Pilot Copies "Stop Turn". If a climbout is in progress for any reason (PTCLO = .true.), the pilot disregards this approach event. Otherwise the pilot's concept of the current "correct" rate-of-climb, rate-of-turn, and airspeed changes as follows:

Airspeed (PTASM)

Unaffected

Rate-of-Climb (PTYDM)

Unaffected

Rate-of-Turn (PTHDM, radians/second)

Changes to 0.0 degrees/second.

No Radio Contact. The simulated GCA-controller's "microphone" does not change its "key-state," nor does the GCA-controller transmit an advisory (neither "loud-and-clear" nor "weak-but-clear") for a period of PTMAXNCC half-seconds. Pilot "thought process" reacts as under "Pilot Copies Execute Missed Approach") below.

Pilot Copies "Low Altitude Alert". If the APE stated variable PTGPADSBL is set to ".true." by the user via the courseware, the actual rate-of-climb for the simulated aircraft is fixed at the "minimum admissible rate-of-climb", PTMINYDI (which equals the standard no-wind ideal rate-of-descent minus 500 feet/minute), in order to induce a "low-altitude alert" within range ACLOW feet from touchdown.

Upon encountering this approach event, the pilot's concept of current "correct" rate-of-climb, rate-of-turn, and airspeed changes as follows:

Airspeed (PTASM)

Unaffected

Rate-of-Turn (PTHDM)

Unaffected

Rate-of-Climb (PTYDM, feet/second)

Changes to +500 feet/minute.

Pilot Copies "Execute Missed Approach" or "Missed Approach". Pilot's concept of the "correct" current rate-of-climb, rate-of-turn, and airspeed changes as follows:

a. If aircraft altitude is greater than or equal to 1500 feet when "approach event" occurs:

Airspeed (PTASM feet/second)

"Correct" airspeed (knots) is "Pattern Airspeed" for aircraft type (ACTYP) being simulated, as shown in Table 19.

TABLE 19. PATTERN AIRSPEEDS

Aircraft Type (ACTYP)	Pattern Airspeed (knots)*
1	120
2	140
3	160
4	180

* = SPTAS (ACTYP), = PTASPAT, feet/second

Rate-of-Turn (PTHDM, radians/second)

"Correct" rate-of-turn is +3.0 degrees/second until reaching a heading of 300°, i.e., PTHMTN ← 300°. Then "correct" rate-of-turn becomes 0.0 degrees/second.

Rate-of-Climb (PTYDM, feet/second)

Becomes 0.0 feet/minute.

b. If aircraft altitude is less than 1500 feet when this "approach event" occurs:

Airspeed (PTASM, feet/second)

"Correct" airspeed (knots) is "Climbout Airspeed" for aircraft type being simulated, as shown in Table 20.

TABLE 20. CLIMBOUT AIRSPEEDS

Aircraft Type (ACTYP)	"Climbout Airspeed" (knots)*
1	150
2	175
3	195
4	235

* = SCOAS (ACTYP), = PTASCLO, feet/second

When aircraft reaches an altitude of 1500 feet, (i.e., PTYMTN ← 1500), the "correct" airspeed becomes the same as "a" above.

Rate-of-Turn (PTHDM, radians/second)

"Correct" rate-of-turn becomes 0.0 degrees/second until aircraft reaches an altitude of 1500 feet (i.e., PTYMTN ← 1500); then "correct" rate-of-turn becomes the same as "a" above.

Rate-of-Climb (PTYDM, feet/second)

"Correct" rate-of-climb is "Climbout Rate of Climb" for aircraft type being simulated, as shown in Table 21.

TABLE 21. CLIMBOUT RATE OF CLIMB

Aircraft Type (ACTYP)	Climbout Rate of Climb (feet/minute)*
1	1000
2	4000
3	1000
4	4000

* = SCOYD (ACTYP), = PTYDCLO, feet/second

When aircraft reaches an altitude of 1500 feet (i.e., PTYMTN ← 1500), the "correct" rate of climb becomes the same as "a" above.

Pilot Copies "If Runway Not in Sight, Execute Missed Approach". If the current aircraft altitude exceeds the user-specified simulated ceiling height (ENCEIL), the pilot disregards this approach event; otherwise the pilot's "thought process" reacts as in subsection "Pilot Copies 'Execute Missed Approach' or 'Missed Approach'" above.

Pilot Copies "If Runway Not in Sight, Climb and Maintain 1500". Same as for the previous subsection, "Pilot Copies 'If Runway Not in sight, Execute Missed Approach'," above except that the pilot's concept of the current "correct" rate-of-turn becomes 0.0 degrees/second and remains such until changed upon occurrence of some subsequent "approach event," if any.

Pilot Copies "Proceed Direct Point Bravo". Same as for "Pilot Copies 'Execute Missed Approach'" above, except for substituting "3000 feet" for "1500 feet" and "270°" for "300°" (i.e., PTYMTN ← 3000, PTHMTN ← 270°).

Pilot Copies "Climb and Maintain 1500". Same as for subsection "Pilot Copies 'If Runway Not in Sight, Climb and Maintain 1500'", except that the maneuver is executed even if the current aircraft altitude exceeds the ceiling height.

Pilot Copies "Climb and Maintain 3000". Same as for subsection "Pilot Copies 'Climb and Maintain 1500'" above, except for substituting "3000 feet" for "1500 feet" (i.e., PTYMTN ← 3000).

Pilot Assumes Instrument Failure Has Occurred. If at any time during the approach the pilot's current estimate of the "true descent-parallel-to-glidepath rate-of-climb", $\dot{y}_p(t)$, falls outside the range P-500 to P+200 (feet/minute; P = PTSYDI, feet/second, = "Standard Initial Final Approach Rate-of-Climb"; P-500 feet/minute denoted PTMINYDI, feet/second; P+200 feet/minute denoted PTMAXYDI, feet/second), then the pilot's concept of the current "correct" rate-of-turn, rate-of-climb, and airspeed changes as stated in subsection "Pilot Copies 'Execute Missed Approach' or 'Missed Approach'" above.

Pilot Copies "At Decision Height". If any of the following conditions holds, the pilot's concept of the current "correct" rate-of-climb, rate-of-turn, and airspeed changes as in subsection "Pilot Copies 'Execute Missed Approach' or 'Missed Approach'" above; otherwise the pilot disregards this approach event.

a. Approach type being simulated is a "low approach" (PTAPR=2).

b. Approch type being simulated is "touch and go" (PTAPR=3) and pilot has copied "cleared for touch and go" at some earlier time during the approach.

c. Approach type being simulated is "full stop" (PTAPR=1), "no-gyro" (PTAPR=5), or "minimum fuel" (PTAPR=4), and pilot has copied "cleared to land" at some earlier time during the approach.

Aircraft Reaches/Passes-Through "Assigned Altitude" on Climbout. Associated with each climbout executed in response to a controller advisory or other approach event is an assigned altitude, PTYMTN, which the pilot attempts to reach and then maintain. When MOVEPILOT detects that the aircraft's current altitude equals or exceeds the current assigned altitude, the pilot's concept of the current "correct" rate-of-climb, rate-of-turn, and airspeed is changed as follows:

Rate-of-Climb (PTYDM, feet/second)

Becomes 0.0 feet/minute. (The aircraft altitude, ACY, is set to assigned altitude PTYMTN simultaneously; the actual rate-of-climb, ACYD, is set to PTYDM simultaneously).

Rate-of-Turn (PTHDM, radians/second)

Unaffected, unless (1) pilot copied "Turn Left/Right Heading XXX" during climbout, in which case "correct" rate-of-turn becomes as subsection "Pilot Copies 'Turn Left/Right Heading XXX'" with Left/Right and XXX interpreted according to the most recent turn advisory copied during the current climbout, if any (i.e., PTHMTN ← XXX); or (2) climbout resulted from occurrence of the "Missed Approach" events above and there have been no subsequent turn advisories copied, in which case the "correct" rate-of-turn becomes as described in the appropriate "Missed Approach" subsection (i.e., PTHMTN ← PTWMAH).

Airspeed (PTASM, feet/second)

Becomes "Pattern Airspeed" corresponding to aircraft type (ACTYP) being simulated (see previous subsection "Pilot Copies 'Execute Missed Approach' or 'Missed Approach'").

Aircraft Reaches/Passes Through "Assigned Heading" When Turning. If the current actual aircraft heading, ACH, lies within 2° of the current assigned heading, PTHMTN, while the aircraft is turning (PTTRH = .true. or PTTLH = .true.) with an operational gyro (ACGYRO = .true.), then the aircraft's actual heading (ACH) is immediately set to PTHMTN, the actual rate-of-turn, ACH, is set to 0.0 degrees/second, and the pilot's concept of the "correct" rate-of-climb, rate-of-turn, and airspeed changes as follows:

Airspeed (PTASM)

Unaffected

Rate-of-Climb (PTYDM)

Unaffected

Rate-of-Turn (PTHDM, radians/second)

Becomes 0.0 degrees/second.

Pilot Acquires Visual Contact With Runway. If the current aircraft altitude (ACY) is less than the user-specified simulated ceiling height (ENCEIL), the pilot no longer maintains a concept of the "correct" current rate-of-climb, rate-of-turn, or airspeed for his aircraft, as described in the preceding (and subsequent) sections. APE assumes that once the runway is visually acquired, the pilot will maneuver his aircraft such that its track coincides with a straight line from its current position through the touchdown point.

The variables PTASM, PTYDM, and PTHDM are subsequently ignored, and direct substitution of the values

$$\dot{x}(t+\Delta t) = p\cdot x(t)$$

$$\dot{y}(t+\Delta t) = p\cdot y(t) \qquad p = \frac{V}{\sqrt{x(t)^2+y(t)^2+z(t)^2}}$$

$$\dot{z}(t+\Delta t) = p\cdot z(t)$$

where V = final approach airspeed, = PTASFA

is made into the aircraft dynamics algorithm described under the aircraft performance sections.

THE PILOT/AIRCRAFT PERFORMANCE MODEL. At any given time during a real GCA the pilot's control behavior may be characterized as an attempt to achieve and maintain that unique rate-of-turn/rate-of-climb/airspeed which the pilot judges to be the "correct" one (i.e., the one consistent with the "rules for pilots conducting GCAs") in the light of whatever sequence of "approach events" has transpired up until that time. Previous sections have described the manner in which APE simulates (1) the detection of "approach events" and (2) the process by which the pilot formulates his concept of the current "correct" rate-of-turn/rate-of-climb/airspeed which he then should (and does) attempt to achieve and maintain.

The motion of a real aircraft on GCA, however, is clearly not a function of its pilot's idea of the "correct" rate-of-turn/rate-of-climb/airspeed for the aircraft at each moment; it is a function of the actual rates-of-turn/rates-of-climb/airspeeds which the pilot's control behavior succeeds in eliciting from his aircraft. Clearly, too, the degree to which a pilot succeeds in eliciting desired flight performance from his aircraft depends both on the pilot's skill level and on the handling characteristics of his aircraft. APE simulates the imperfect ability of a real pilot to elicit precisely the desired rate-of-turn/rate-of-climb/airspeed performance from a given aircraft in the MOVEPILOT routine, as shown in Figure 42. This is accomplished by representing, in MOVEPILOT, the "actual" rate-of-turn (ACHD), rate-of-climb (ACYD), and airspeed (ACAS) of the simualted aircraft as random-valued functions of the simulated pilot's "intended" ("correct") rate-of-turn (PTHDM), rate-of-climb (PTYDM), and airspeed (PTASM), his skill level (PTYP), and certain other parameters (see below) which guarantee that the probability density functions (pdf) of the random-valued functions will conform to certain known aircraft/pilot performance statistics.

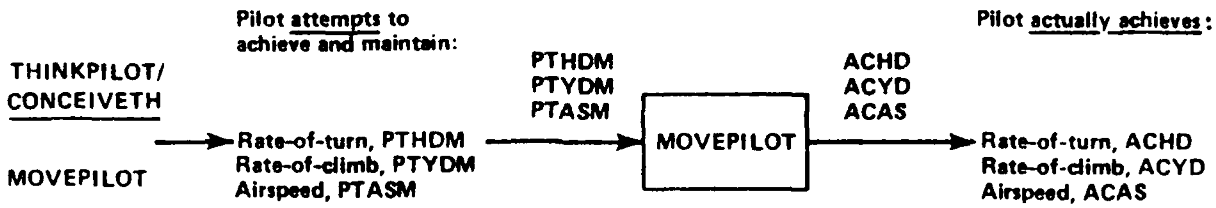


Figure 42. The Function of MOVEPILOT

The general form of these random-valued functions is as follows:

$$v_{actual}(t+0.5) = \alpha\left[v_{actual}(t)\right] + (1-\alpha)\left[kR+(\underset{(\text{"correct"})}{v_{intended}}(t)+\Delta\mu)\right]$$

where

t

is the simulated time (elapsed seconds), and

$v_{actual}(t)$

is the rate-of-turn/rate-of-climb/airspeed (ACHD/ACYD/ACAS) that the simulated pilot actually achieves at time t. This is the value which is used later in MOVEPILOT to compute the simulated aircraft position and heading.

$v_{actual}(t + 0.5)$

is the rate-of-turn/rate-of-climb/airspeed (NEWHD/NEWYD/NEWAS) that the simulated pilot actually achieves at time t + 0.5. This value is used later in MOVEPILOT to compute the simulated aircraft position and heading.

α

is the rate-of-turn/rate-of-climb/airspeed "stability" parameter (PTA1HD/PTA1YD/PTA1AS), where $0 \leq \alpha \leq 1$ and $(1-\alpha)$ is denoted (PTA2HD/PTA2YD/PTA2AS). The greater the value of α, the lower the mean-crossing frequency of rate-of-turn/rate-of-climb/airspeed, the greater the tendency of the aircraft/pilot to precisely maintain a desired rate-of-turn/rate-of-climb/airspeed once it is achieved, and the more "sluggish" the aircraft/pilot response when moving from one desired rate-of-turn/rate-of-climb/airspeed to a subsequent one.

α is assumed to vary inversely with pilot skill level.

R

is a pseudo-normally distributed random variable ($R = X_1+X_2$ where X_1 and X_2 are uncorrelated random variables uniformly distributed over [-32767, 32767]) whose values fall within the range $-65534 \leq R \leq 65534$ and whose mean value is 0.

$V_{\text{intended ("correct")}}(t)$

is the value of rate-of-turn/rate-of-climb/airspeed, PTHDM/PTYDM/PTASM which the simulated pilot is attempting to achieve at time t. This is the "correct" value of rate-of-turn/rate-of-climb/airspeed which the pilot formulates in THINKPILOT/CONCEIVETH and MOVEPILOT.

$k, \Delta\mu$

are the rate-of-turn/rate-of-climb/airspeed pdf parameters, PTKHD, PTMHD/PTKYD, PTMYD/PTKAS, PTMAS, which are automatically selected by APE such that the resulting pdf of "actual" rate-of-turn, (ACHD/NEWHD) rate-of-climb (ACYD/NEWYD/) Airspeed (ACAS/NEWAS) will have the characteristics embodied in the graph shown in Figure 43.

The values of the various pdf parameters currently coded into APE correspond to the pilot profiles shown in Table 22.

TABLE 22. PILOT PROFILES

Pilot Type	Pilot maintains desired rate-of-turn within a range of: (degrees/second) (See Figure 43) λ_e λ_n	Pilot maintains desired rate-of-climb within a range of: (feet/minute) (See Figure 43) λ_e λ_n	Pilot maintains desired Airspeed within a range of: (knots) (See Figure 43) λ_e λ_n
1	-.5 to +.5	-50 to +50	-5 to +5
2	-1.0 to +1.0	-75 to +75	-5 to +10
3	-1.5 to +1.5	-100 to +100	-5 to +15
4	-2.0 to +2.0	-125 to +125	-5 to +20
5	-3.0 to +3.0	-150 to +150	-5 to +25

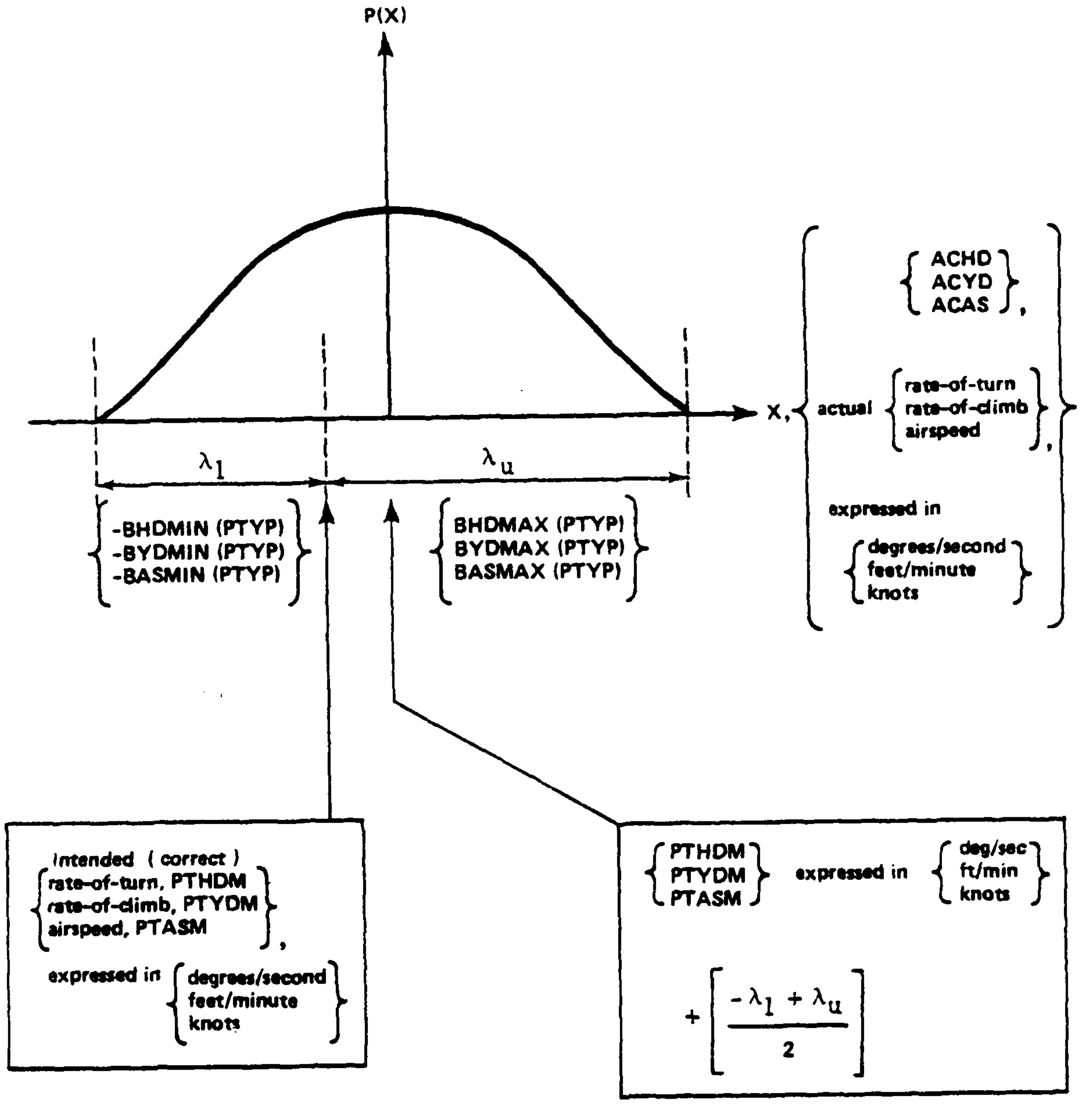


Figure 43. Distribution of "pdf" Parameters

Once MOVEPILOT has determined NEWHD, NEWYD, and NEWAS (the "actual" values of rate-of-turn, rate-of-climb, and airspeed to be simulated at time t + 0.5), the computations which determine the "next" aircraft position, (x[t+0.5], y[t+0.5], z[t+0.5]) are elementary, as shown graphically in Figure 44.

Given:

t,	= current time (seconds elapsed)
Δt,	= 0.5 seconds
$\dot{h}(t+\Delta t)$,	= the "actual" rate-of-turn at time $t+\Delta t$ (radians/second)
$\dot{y}(t+\Delta t)$,	= the "actual" rate-of-climb at time $t+\Delta t$ (feet/second)
$s_a(t+\Delta t)$,	= the "actual" airspeed at time t+ t (feet/second)

Let $h(t)$ = aircraft heading at time t (radians)

Then

$$h(t+\Delta t) = h(t) + \int_t^{t+\Delta t} \dot{h}(t)dt \cong \left[h(t) + \left(\frac{\dot{h}(t+\Delta t) + \dot{h}(t)}{2} \right) \Delta t \right] \bmod 2\pi$$

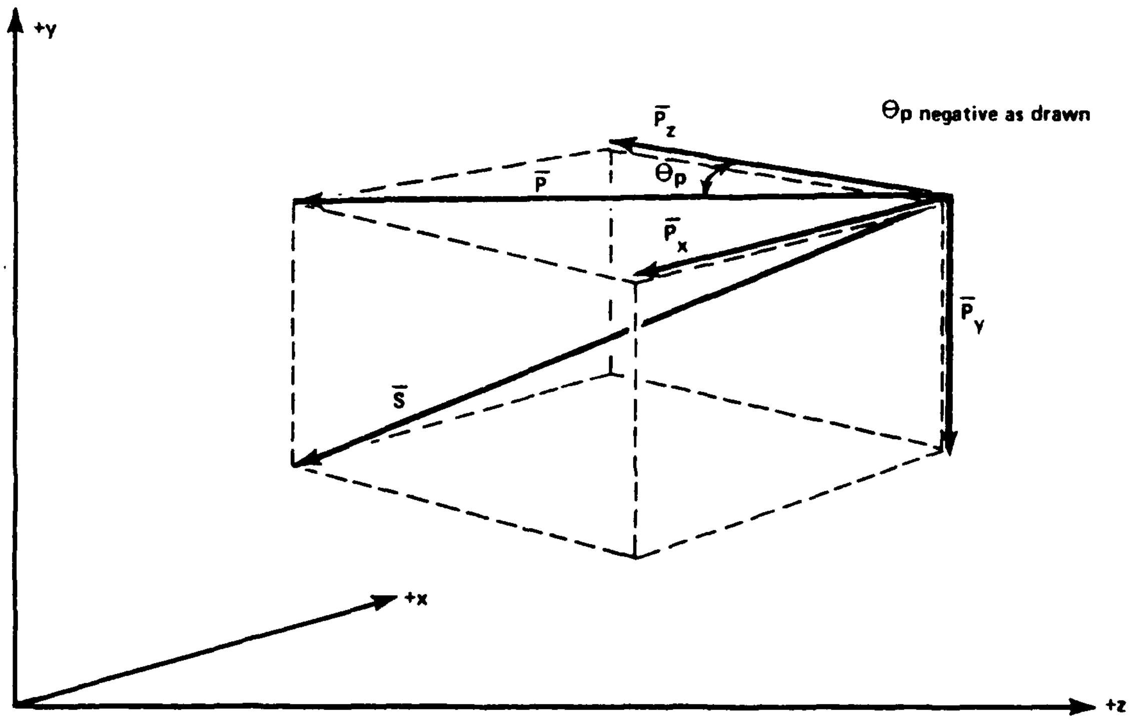


Figure 44. MOVEPILOT Determination of Next Aircraft Position

Next let $\vec{S} = \vec{P}_x + \vec{P}_y + \vec{P}_z$ be the aircraft velocity at time $t+\Delta t$ with respect to the frame of reference of the surrounding airmass, where $\vec{P}_x$, $\vec{P}_y$ and $\vec{P}_z$ are the x, y, and z components of $\vec{S}$, respectively.

Let $\vec{P} = \vec{P}_x + \vec{P}_z$, then $\vec{S} = \vec{P} + \vec{P}_y$

and $|\vec{S}|^2 = |\vec{P}|^2 - |\vec{P}_y|^2$

But $|\vec{S}| = Sa(t+\Delta t)$ and $|\vec{P}| = \dot{y}(t+\Delta t)$, so

$$|\vec{P}| = \sqrt{[Sa(t+\Delta t)]^2 - [\dot{y}(t+\Delta t)]^2}$$

(See Figure 43.)

Since $\Theta p = h(t+\Delta t) - (160 \cdot \frac{2\pi}{360})$

Then:

$$|\vec{Pz}| = -|\vec{P}|\cos\left(h(t+\Delta t) - \frac{8\pi}{9}\right) = -\left(\sqrt{[Sa(t+\Delta t)]^2 - [\dot{y}(t+\Delta t)]^2}\right)\cos\left(h(t+\Delta t) - \frac{8\pi}{9}\right)$$

and $$|\vec{Px}| = |\vec{P}|\sin\left(h(t+\Delta t) - \frac{8\pi}{9}\right) = \left(\sqrt{[Sa(t+\Delta t)]^2 - [\dot{y}(t+\Delta t)]^2}\right)\sin\left(h(t+\Delta t) - \frac{8\pi}{9}\right)$$

So if $\dot{x}(t)$, $\dot{y}(t)$, $\dot{z}(t)$ are the x, y, and z components of aircraft velocity at time t with respect to the simulation co-ordinate system ("ground") frame of reference, then:

$$\dot{x}(t+\Delta t) = |\vec{P}_x| + W_x(t+\Delta t)$$

$$\dot{y}(t+\Delta t) = |\vec{P}_y|$$

$$\dot{z}(t+\Delta t) = |\vec{P}_z| + W_z(t+\Delta t)$$

where $W_x(t+\Delta t)$, $W_z(t+\Delta t)$ are the x and z components of wind velocity at time $t+\Delta t$ with respect to the "ground" frame of reference.

We may now compute the simulated aircraft position at time $t+\Delta t$ as follows:

$$x(t+\Delta t) = \int_t^{t+\Delta t} \dot{x}(t)\,dt \cong x(t) + \left(\frac{\dot{x}(t+\Delta t) + \dot{x}(t)}{2}\right)\Delta t$$

$$y(t+\Delta t) = \int_t^{t+\Delta t} \dot{y}(t)\,dt \cong y(t) + \left(\frac{\dot{y}(t+\Delta t) + \dot{y}(t)}{2}\right)\Delta t$$

$$z(t+\Delta t) = \int_t^{t+\Delta t} \dot{z}(t)\,dt \cong z(t) + \left(\frac{\dot{z}(t+\Delta t) + \dot{z}(t)}{2}\right)\Delta t$$

The correspondence between symbols in the above expression and Fortran identifiers in the APE implementation are shown in Table 23.

RESTRICTED ELEVATION AND AZIMUTH MODE SIMULATION. The preceding sections described the behavior of the GCA-CTS simulated aircraft/pilot/environment when GCA-CTS is operated in "unrestricted mode." GCA-CTS also provides the capability of simulated restricted elevation or azimuth approaches. When GCA-CTS is operated in the restricted mode, the GCA-CTS simulated PAR display exhibits the simulated radar image of an aircraft which, while attempting to execute a GCA, maintains precisely the "correct" horizontal distance from the course/centerline or vertical distance from the glidepath throughout the approach. However, the vertical distance from the glidepath or horizontal distance from the course/centerline oscillates approximately sinusoidally between the upper and lower or leftmost and rightmost extremes of a user-specified set of contiguous "permissible" elevation or azimuth zones (e.g., above-glidepath through well-above-glidepath or on-course through right-of-course) at a constant user-specified frequency. Successive positions of the simulated aircraft are obtained by cyclic execution of subroutines APREX and APRAX. Since routines THINKPILOT/SPEAKPILOT/MOVEPILOT are never invoked, the motion of the simulated aircraft does not vary in response to "approach events" (e.g., simulated GCA advisories), nor does the simulated pilot "speak" during the approach.

Note that GCA-CTS may be operated in either elevation- or azimuth-restricted mode, but not in both modes simultaneously; i.e., the simulated aircraft cannot be made to oscillate "sinusoidally" in the x-z and the y-z planes concurrently.

The value of ACZVF hard-coded into the current version of APE was selected to impart to the simulated aircraft a constant z-axis velocity component of 120 knots. Since either the x- or y-axis velocity components is time-varying whenever the other is constant, the net velocity of the simulated aircraft is always "sinusoidally" time-varying in either restricted mode.

The value of SECPZ hard-coded into the current version of APE causes the simulated aircraft to traverse the user-specified set of contiguous elevation or azimuth zones in $5W_e$ or $5W_a$ seconds, one-way, extremum to extremum. The amount of time that the aircraft resides in a given zone during a single traverse is a non-constant function of the zone number and W_e or W_a.

Restricted Elevation. The motion of the simulated aircraft in the restricted elevation mode is shown in Figure 45 and may be described parametrically as follows (t = time, elapsed half-seconds):

$$x(t) = 0$$

$$y(t) = \mu(t) + \alpha(t)\sin\left[k_e(Z_o - Z(t))\right] = \mu(t) + \alpha(t)\sin(kvt)$$

$$z(t) = Z_o - vt$$

TABLE 23. CORRESPONDENCE BETWEEN SYMBOLS USED IN THIS DISCUSSION AND APE FORTRAN IDENTIFIERS

Symbol	FORTRAN Identifier
$\ddot{h}(t)$	ACHD
$\dot{y}(t)$	ACYD
$S_a(t)$	ACAS
$h(t)$	ACH
Δt	coded as a constant; "0.5,"
$\ddot{h}(t+\Delta t)$	NEWHD
$\dot{y}(t+\Delta t)$	NEWYD
$S_a(t+\Delta t)$	NEWAS
$h(t+\Delta t)$	NEWH
$\lvert\vec{P}\rvert$	NEWXZS
$\sin\left(h(t+\Delta t)-\frac{8\pi}{9}\right)$	NEWHX
$\cos\left(h(t+\Delta t)-\frac{8\pi}{9}\right)$	NEWHZ
$\dot{x}(t)$	ACXD
$\dot{z}(t)$	ACZD
$\dot{x}(t+\Delta t)$	NEWXD
$\dot{y}(t+\Delta t)$	NEWYD
$\dot{z}(t+\Delta t)$	NEWZD
$x(t)$	ACX
$y(t)$	ACY
$z(t)$	ACZ
$x(t+\Delta t)$	NEWX
$y(t+\Delta t)$	NEWY
$z(t+\Delta t)$	NEWZ

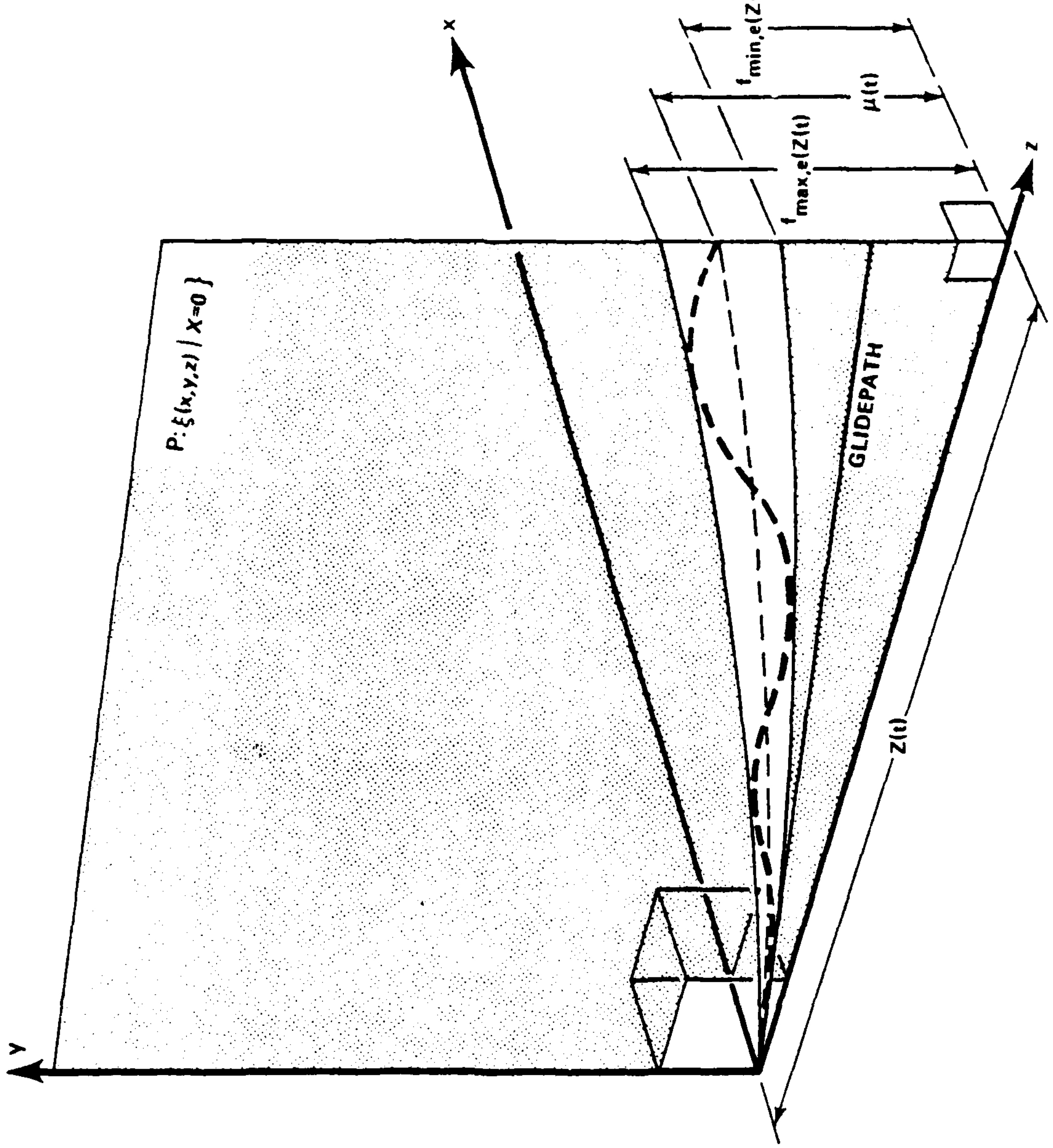


Figure 45. Restricted Elevation Flight Geometry

where:

$$\mu(t) = \frac{f_{max,e}\Big(z(t)\Big) + f_{min,e}\Big(z(t)\Big)}{2}$$ "center of oscillation" function

$$a(t) = f_{max,e}\Big(z(t)\Big) - f_{min,e}\Big(z(t)\Big)$$ amplitude of oscillation

$$k_e = \frac{\left(\frac{\pi}{2}\right)}{P_e V W_e}$$ frequency of oscillation factor

where:

$f_{max,e}(z[t])$ is the altitude (in feet) corresponding to the upper boundary - at a range of z(t) feet from touchdown - of the user-specified set of contiguous "permissible" elevation zones.

$f_{min,e}(z[t])$ is the altitude (in feet) corresponding to the lower boundary - at a range z(t) feet from touchdown - of the user-specified set of contiguous "permissible" elevation zones.

P_e is the average number of seconds the aircraft resides in each zone of the user-specified set of contiguous "permissible" elevation zones during each (one-way) traverse of the set.

V is the z-axis velocity of the simulated aircraft expressed in feet/half-second.

W_e is the width (in number of zones) of the user-specified set of contiguous "permissible" elevation zones.

Z_o is the user-specified starting range (feet). The approach begins with the aircraft at $(0, \mu(0), Z_o)$.

The correspondences between symbols in the above expressions and Fortran identifiers in subroutines APPREX, APENIT, and APESNIT are shown in Table 24.

TABLE 24. CORRESPONDENCE BETWEEN SYMBOLS USED IN RESTRICTED ELEVATION DISCUSSION AND FORTRAN IDENTIFIERS

Symbol	FORTRAN Identifier
$x(t)$	ACX
$y(t)$	ACY
$z(t)$	ACZ
$f_{max,e}(z[t])$	FMAX
$f_{min,e}(z[t])$	FMIN
k_e	ACQF
P_e	SECPZ
V	ACZVF
W_e	PLTUZN - PLTLZN + 1
Z_o	ACZ0

Restricted Azimuth. The motion of the simulated aircraft in the restricted azimuth mode is shown in Figure 46 and may be described parametrically as follows. Again t = time, elapsed half-seconds.

$$x(t) = \mu(t) + \alpha(t)\sin\left[k\left(z_o - z(t)\right)\right] = \mu(t) + \alpha(t)\sin(kvt)$$

$$y(t) = (0.0524)z(t) = z(t)(\tan 3°)$$

$$z(t) = z_o - vt$$

$$\mu(t) = \frac{f_{max,e}\left(z(t)\right) + f_{min,e}\left(z(t)\right)}{2},$$ "center of oscillation" function

$$\alpha(t) = f_{max,e}\left(z(t)\right) - f_{min,e}\left(z(t)\right),$$ amplitude of oscillation

$$k = \frac{\left(\frac{\pi}{2}\right)}{P_e N W_e},$$ frequency of oscillation factor

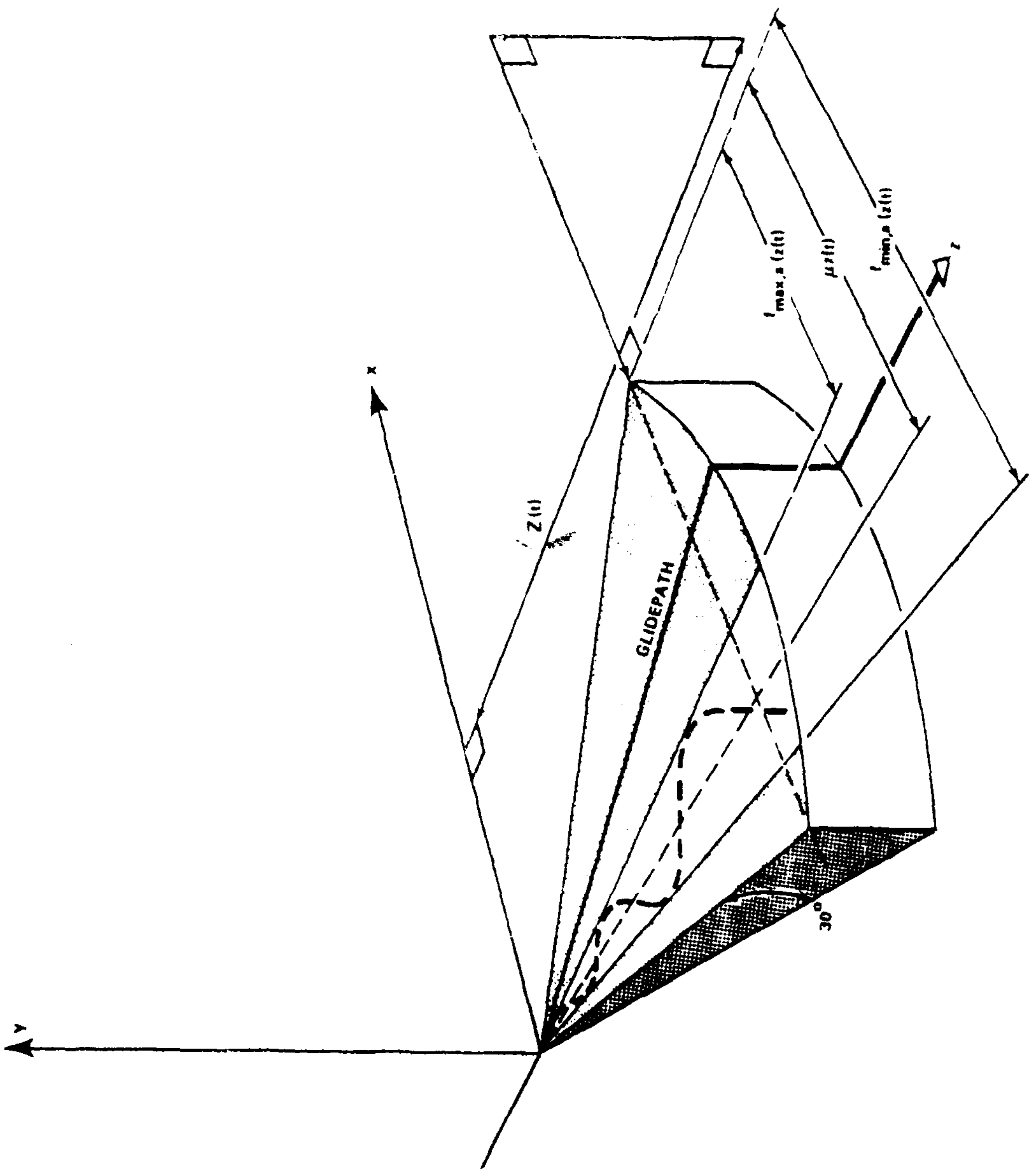


Figure 46. Restricted Azimuth Flight Geometry

where:

$f_{max,a}(z[t])$ is the horizontal distance (x-axis offset), in feet, corresponding to the rightmost boundary - at a range of z(t) feet from touchdown - of the user-specified set of contiguous permissible azimuth zones.

$f_{min,a}(z[t])$ is the horizontal distance in feet corresponding to the leftmost boundary at a range z(t) feet from touchdown of the user-specified set of contiguous permissible azimuth zones.

p_a is the average number of seconds that the aircraft resides in each zone of the user-specified set of contiguous azimuth zones during each one-way traverse of the set.

V same as in Restricted Elevation discussion above.

W_a is the width (in number of zones) of the user-specified set of contiguous permissible azimuth zones.

Z_o same as in Restricted Elevation discussion above.

The correspondences between the symbols in the above expressions and the Fortran identifiers appearing in subroutines APRAX, APENIT, and APESNIT are shown in Table 25.

TABLE 25. CORRESPONDENCE BETWEEN SYMBOLS USED IN RESTRICTED AZIMUTH DISCUSSION AND FORTRAN IDENTIFIERS

Symbol	Fortran Identifier
$x(t)$	ACX
$y(t)$	ACY
$z(t)$	ACZ
$f_{max,a}(z[t])$	FMAX
$f_{min,a}(z[t])$	FMIN
k_a	ACQF
p_a	SECPZ
V	ACZVF
W_a	PLTRZN - PLTLZN + 1
Z_o	ACZ0

RADAR SIMULATION

There are two radars which generate the precision approach radar (PAR) display. One is the elevation radar, whose sweep operates in the vertical plane, and the other is the azimuth radar, whose sweep operates in the horizontal plane. The scope of each radar and its corresponding display area are shown in Figures 47 and 48. Thus an aircraft which is within the elevation sweep area will appear in the upper portion of the display and one which is within the azimuth sweep area will appear in the lower portion of the display. The scan of each radar traverses the limits of the other radar scan. Thus the elevation scan, whose horizontal width is relatively narrow can be moved (servoed) right or left to pick up a target which is right or left of course. Similarly, the azimuth scan can be moved up or down in the vertical plane to pick up a target which is above or below glidepath. The radar simulation accepts data from the aircraft/pilot/environment simulation (APE) and translates the data to coordinates which can be displayed on the Megatek display. The radar display is approximately logarithmic to allow more precise manipulation of the aircraft close to touchdown. The radar simulation resides on CPU 1. Figure 49 is a block diagram of this simulation. The radar consists of two routines: RADAR, which translates the data and LOOKUP, which clips the target to conform to the display area.

TRANSLATION ROUTINE. The routine RADAR in fact performs several tasks. It translates an aircraft in physical space to a point in screen space delimited by the maximum area of the Megatek display. This is accomplished by using data furnished by APE on altitude, range to touchdown and offset from centerline. The routine also saves servo information received from CPU 2 in common. It writes to the disk information on target position in screen coordinates,

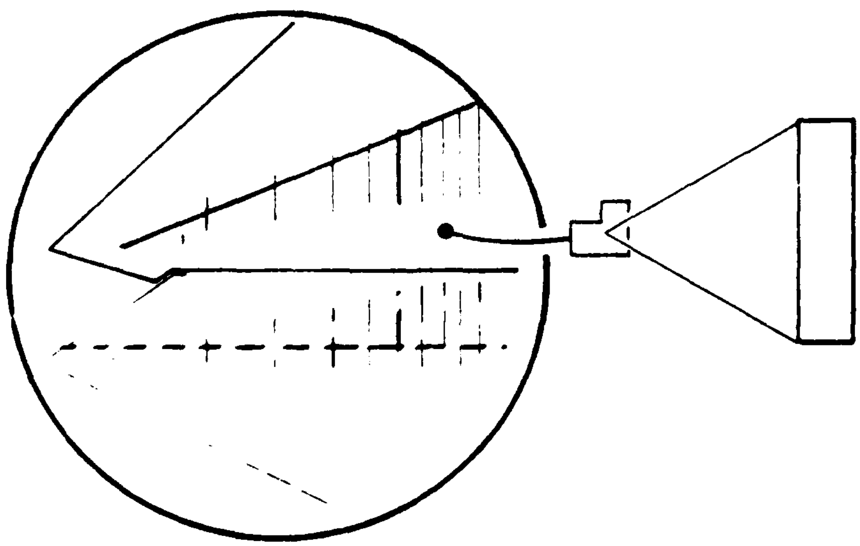


Figure 47. Elevation Radar Sweep

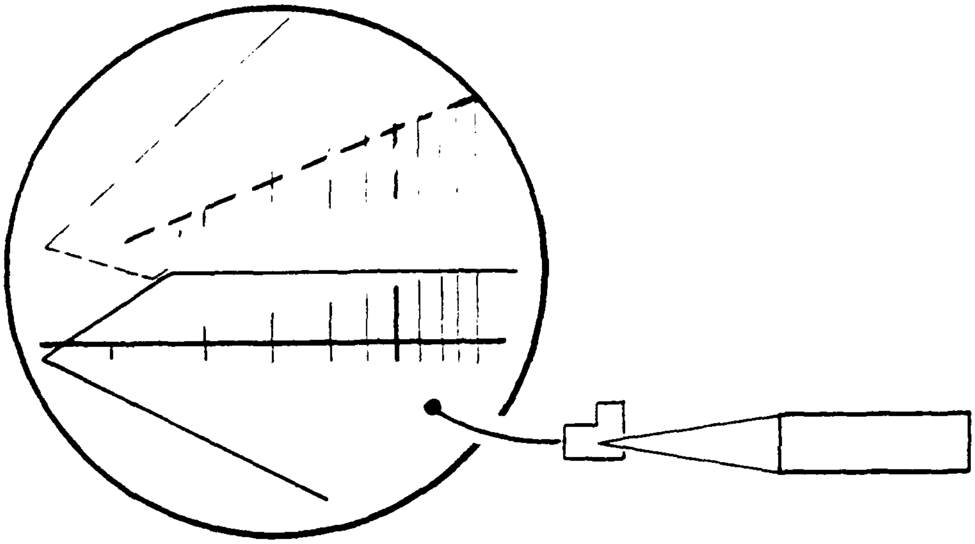


Figure 48. Azimuth Radar Sweep

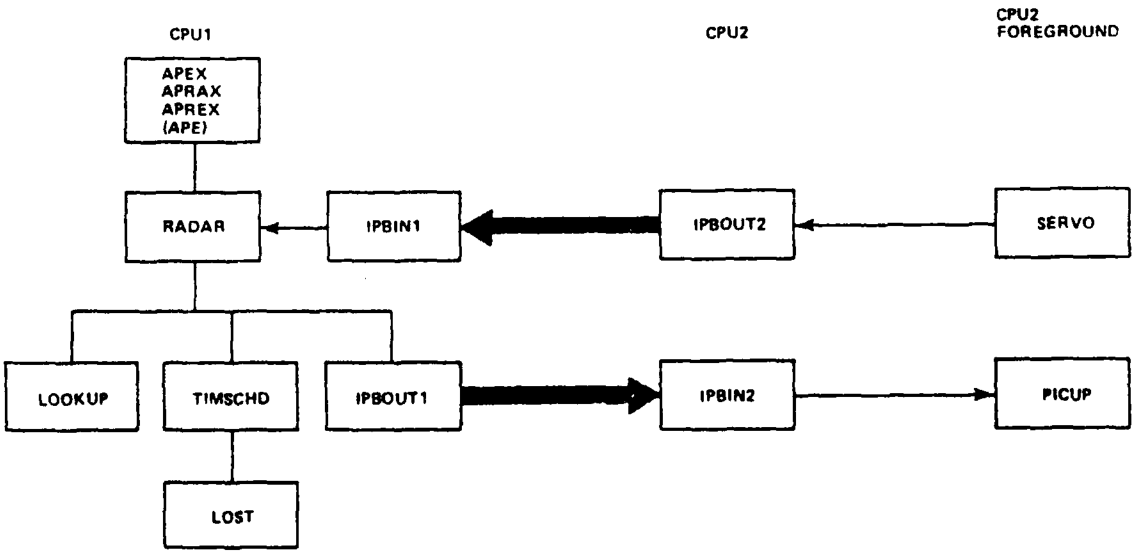


Figure 49. Radar Simulation Block Diagram

servo position, wind speed and heading for use during replay. It determines the end of the physical run defined by the point at which the aircraft has landed, or when the aircraft has disappeared from both display areas due to a missed approach, touch and go, or low approach. For training purposes, the end of a run can also be artificially defined and RADAR will act upon this information. The evaluation as to end-of-run is returned as an argument to APE. Finally RADAR discerns the occurrence of lost radar contact and calls the final controller routine LOST to put the emergency message into controller common.

Display Transformation. In the real GCA environment each point in the vicinity of the glidepath corresponds to two points on the PAR display: one point which is its image on the so-called elevation display portion of the PAR display, and one point which is its image on the so-called azimuth display portion of the PAR display. Figure 50 shows the relationship between the physical situation and the display.

The following procedure is used by GCA-CTS to determine the two such points on the GCA-CTS simulated PAR display for each point (x, y, z) in simulated real-space. (The real-space point (x, y, z) has co-ordinates in feet with respect to the physical coordinate system illustrated in Figure 50. The elevation display and azimuth display images of (x, y, z) are denoted (ζ_E, Y_E) and (ζ_A, Y_A), respectively, with coordinates in Megatek screen units with respect to the display coordinate system illustrated in Figure 50.

Procedure. Given (x, y, z), as described above, the steps in the display transformation procedure are presented as follows:

1. Let $\Delta z \longleftarrow z + 3605.07$

   If $0 \le \Delta z \le 57382$, then continue step 2 below; else (No Point on the simulated display corresponds to (x, y, z); (x, y, z) is either off-screen" to the right, or is behind the real-space position of the simulated PAR, so...) STOP.

$$\zeta_\circ \longleftarrow \left[\left(\frac{4151.61z}{z + 27483.5}\right) - 1053.8\right]$$

$$\zeta_E \longleftarrow \zeta_A \longleftarrow \zeta_\circ$$

$$\zeta^* \longleftarrow \zeta^\circ + 1655.2$$

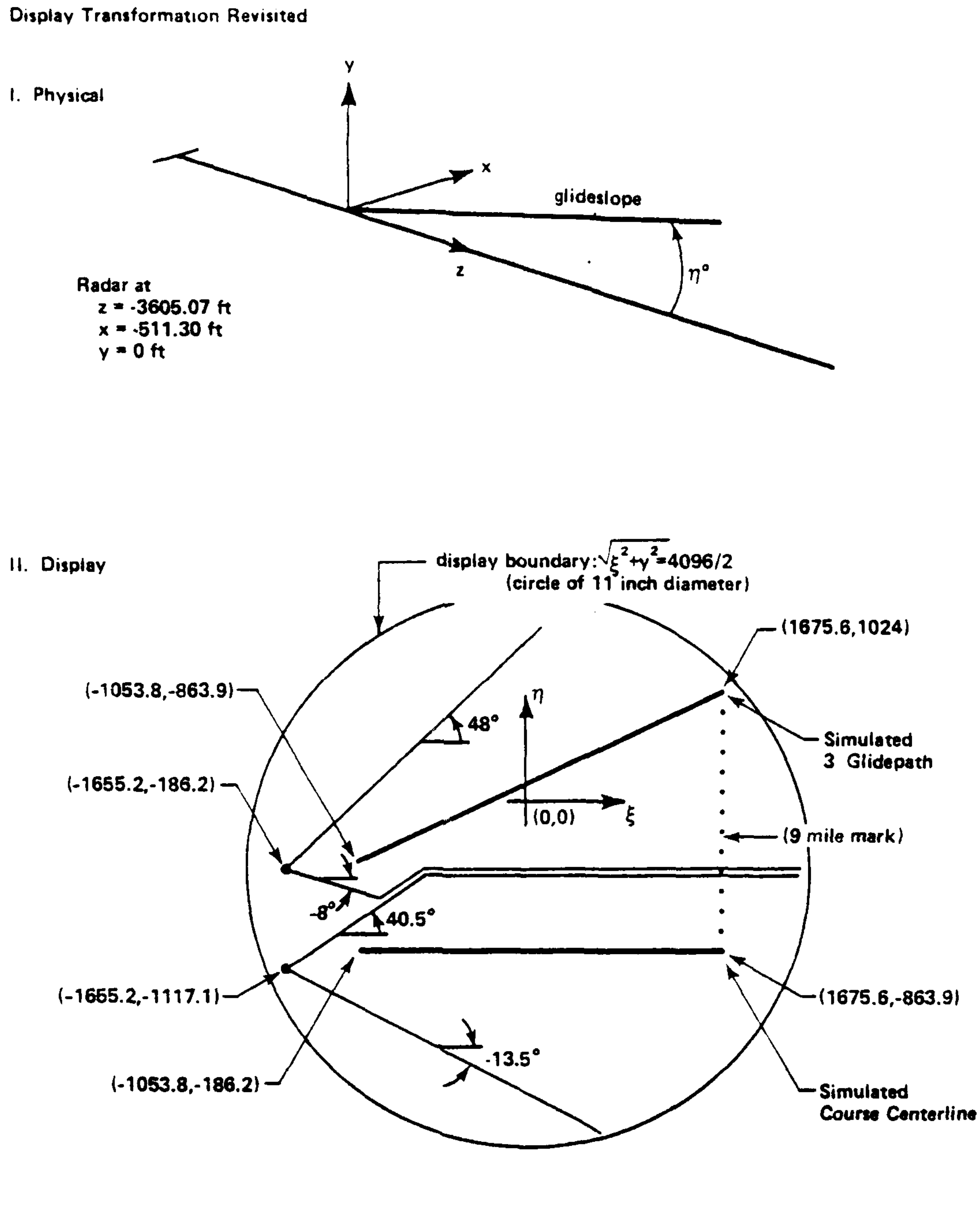


Figure 50. Relation Between Physical Situation (Used in APE) and Display

3. $h_e \longleftarrow \frac{y}{\Delta z}$

If $-.019018 \leq h_e \leq .105104$ then continue to step 4 below; else (No Point on the "elevation display" corresponds to (x, y, z) since (x, y, z) does not lie within the elevation limits of the simulated PAR beam, so...) continue to step 6.

4. $h_d \longleftarrow 7.38983\ h_e$

$\Delta h_e \longleftarrow h_e - .069927$

If $\Delta h_e > 0$ then $h_d \longleftarrow h_d + \left[(269.2839)(\Delta h_e)^2\right]$

5. $\eta_E \longleftarrow (h_d \zeta^*) - 186.2$

6. $h_d \longleftarrow \left[2.96875\left(\frac{x}{\Delta z}\right)\right] + \left(\frac{253.2}{\zeta^*}\right)$

If $-.240079 \leq h_d \leq .85408$ then continue to step 7 below; else (No Point on the "azimuth display" corresponds to (x, y, z) since (x, y, z) does not lie within the azimuth limits of the simulated PAR beam, so...) STOP.

7. $\eta_A \longleftarrow (h_d \zeta^*) - 1117.1$

Properties. This display transformation has the following properties:

a. The displayed point has the same horizontal position in both azimuth and elevation displays.

b. The glideslope is a straight line in both displays.

c. In the elevation display straight lines (physical space) through the radar are displayed as straight lines through the radar position (-1655.2, -186.2) on the display.

d. In the elevation display, the slope of lines which intersect the glideslope between 0 and 9 nautical miles is magnified by the constant 7.38983. (This constant, and the physical z coordinate of the radar, were selected to give good fit to the location of range lines when property e is imposed.)

e. The logarithmic range scale, i.e., the relation between ζ and z, was determined by requiring the glideslope to appear as a straight line, while retaining a constant ratio between the slope of lines in physical space and the slope of lines in the elevation display. (In the procedure the variables h_e and h_a are slopes of lines in physical space and h_d is the slope of the line in display space.)

f. The procedure includes checks to prevent display above and below the display scan limits (+48°, -8° for elevation and +40.5°, -13.5° for azimuth).

g. In elevation, since straight lines through the radar are displayed as straight lines, the display scan limits correspond to radar elevation limits independent of range. The upper limit is exactly 6° (obtained by the quadratic correction to h_d in step 4). The lower limit is $\tan^{-1}(-.019018) = -1.095°$.

h. The elevation display is thus characterized by the radar position in physical space and display space, and the requirements that the glideslope be a straight line, and that there be a linear relation between the physical-space and display-space slopes of lines through the radar for such lines as have slope < 4° in physical space; thereafter it is quadratic (but with continuous variation of scale factor) so as to display +6° at +48°.

i. In azimuth it is not possible to use the same (ζ - z) relation as in elevation and also (i) choose arbitrarily the displayed radar position, (ii) have the course centerline be a straight line, and (iii) have lines through the radar displayed as straight lines. Item (iii) was dropped in order to preserve (i) and (ii). (Imposing (ii) and (iii) led to a very peculiar looking position for the radar. Thus (i) and (ii) were chosen over (ii) and (iii).) Thus straight lines through the radar are not displayed as straight lines in the azimuth display.

j. The azimuth display can be characterized as having slopes of lines in physical space (relative to the line parallel to the runway center, through the radar) amplified by the factor 2.96875 for display, using the (ζ - z) relation established in the elevation display, with an arbitrary function of ζ (or z) added to the y coordinate so as to make the course centerline appear as a straight line.

k. Since straight lines through the radar are not displayed as straight lines in the azimuth display, the azimuth display scan limits (at +40.5° and -13.5°) do not correspond to constant physical azimuth scan limits, but rather depend upon the range. The effective limits are near +15° and -5°, and the factor 2.96875 was chosen to achieve this result.

Aircraft Blipsize. The size of the blip caused by the aircraft is an artifact of the finite radar beamwidth, according to recently obtained information. A simple model for the size of the blip, based on this principle, would indicate that if the aircraft is at an elevation angle ϵ, then the blip appears (a radar signal is detected) when the radar antenna is directed at any elevation between $\epsilon - \beta$ and $\epsilon + \beta$, where β is the antenna's effective half beamwidth. (This model neglects the fourth power variation of radar signal detectability with range, but still may be reasonable.)

In the elevation plane, the upper scan limit of 6° indicates that small angle approximations are reasonable. The radar signal causing target display can therefore be assured to occur when the slope of the radar beam centerline is within a constant deviation of the slope of the line to the aircraft. In view of the constant ratio assured to exist between slopes of lines (through the radar) in physical and display space, this indicates that the aircraft should appear on the elevation display between two lines where slopes differ from the slope of the line to the radar by a simple constant.

The association of blip height with antenna beamwidth establishes blip height as a function of range (z or ζ) within a multiplicative constant. In addition the blip height depends upon aircraft type and a display device adjustment. Specifications indicate that the adjustment is to be made such that an aircraft of specified radar reflectivity (10 square meters cross section) at a specified range (9 nautical miles) should cause blip height of a specified size (1.5 inches). Assuming blip height should be the same in both azimuth and elevation displays, the previously given algorithm for finding display position can easily be modified to yield the upper and lower limits (y upper and y lower) of the blip itself:

to step 2 add

$$\Delta\eta = .04886A\zeta^*$$

(where A is an aircraft type factor, equal to 1.00 for the "standard" aircraft);

to step 5 add (for the elevation display)

$\eta^{e}_{lower} = \eta_E - \Delta\eta$ (the η coordinate of the bottom of the blip)

$\eta^{e}_{upper} = \eta_E + \Delta\eta$ (the η coordinate of the top of the blip);

to step 7 add (for the azimuth display)

$\eta^{a}_{lower} = \eta_A - \Delta\eta$ (the η coordinate of the bottom of the blip)

$\eta^{a}_{upper} = \eta_A + \Delta\eta$ (the η coordinate of the top of the blip)

Additional Display Constraints. The procedure given here confines display on the right to ζ values corresponding to a range (z) of 57382 feet, or about 9.44 nautical miles, and each individual (elevation and azimuth) display to be within the display scan limits. More specifically, the center of the aircraft blip is constrained to lie within these confines. Further constraints must be added to obtain a display picture resembling that on operational equipment with fidelity adequate to support the full training potential of the GCA-CTS. Among the additional considerations are:

a. Separation of the display plane into distinct azimuth, and elevation display areas near the line y = -200.

b. Penetration of display boundaries by the extremities of the display blips.

c. Display coordinates outside the 11-inch display scope boundary.

Range Hashmarks. The ζ coordinates of the range hashmarks were given earlier and are the same for both azimuth and elevation displays. The height and vertical position of these hashmarks on a real PAR display is apparently determined by a manual slewing control. In principle the hashmarks are contained within two straight lines in display space passing through the radar position. Furthermore they are two inches high at the nine-mile mark. If the center of the hashmark sequence has slopes in display space then the hashmark for range r can be described as a vertical line segment at ζ coordinate ζ_r (given earlier), between the limits

$$\eta_{\text{lower}}^{\text{upper}} = (S\pm.223588)\zeta_r - 186.2 \text{ (elevation)}$$

$$(S\pm,223577)\zeta_r - 1117.1 \text{ (azimuth)}$$

Position Error Interpretation. The pilot model portion of APE infers aircraft position from advisories. The model is based in part on the position inferred by an almost omniscient pilot who knows the "rules" the controller should have followed in selecting an advisory corresponding to the position of his blip relative to the displayed glideslope. Supporting data required are the blipsize in feet as a function of position. Interpreting the blipsize model in "physical space," one finds that the blip effectively extends both above and below the aircraft a distance

$$\Delta y_{\frac{6}{2}} = (\Delta z)\left(\frac{1}{7.39}\right)\left(\frac{279.27}{3330.8}\right) A$$

$$= (z + 3605.07)(0.0113456)A \qquad \text{(feet)}$$

and to both the left and the right of the aircraft a distance

$$\Delta x_{\frac{6}{2}} = (\Delta z)\left(\frac{1}{2.97}\right)\left(\frac{279.27}{3330.8}\right) A$$

$$= (z + 3605.07)(0.028231)A \qquad \text{(feet)}$$

(Note: GCA-CTS assumes that slewing will render the displayed blipheight essentially constant for all target types; therefore GCA-CTS treats "A" in the above equations as = 1.0 in all cases.)

CLIPPING ROUTINE. Once the target has been translated to a point in display space it is necessary to expand the point to normal target size and to restrict the target to the servo and/or radar display area. This restriction is more complex than is immediately apparent because the location of the hashmarks on the azimuth display affects the clipping of the elevation target and vice versa. The logic used to cause this phenomenon is in the routine called LOOKUP. For ease of explanation, two terms should be defined.

a. The total area is the area in which a target may be seen. It is defined in terms of the limits of the sweep on azimuth and elevation.

b. The display area is the area in which the target could presently be seen. It is defined by either the total area, or the area indicated by the slope of the servo hashmarks to the point at which the sweep originates. There are two display areas: one for azimuth, one for elevation.

To solve the clipping problem previously mentioned, LOOKUP calculates what the target size would be if the target fit entirely within the display area. This target size is based on the range, as in a normal PAR display. The target is compared temporarily with the display area within which it is apparently contained. If it would be clipped by the display area, the amount by which it would be clipped is subtracted from the target in the other display area and the new target is centered around the aircraft point. More explicitly, the clipping is accomplished as follows. After the normal target size has been calculated, dependent on range, the display area is defined for the active run. This display area is dependent on which servos, if any, are activated. For example, during a demonstration, both servos are frozen, and the total area is used to control target clipping. This is necessary because the final controller model would otherwise have to constantly adjust the servo to keep the target in sight, a time-consuming problem. If the trainee is executing an approach, however, the servos in either the azimuth or elevation display, or both can be activated, reducing the display area to the area defined by the servo hashmarks. This is to give the trainee experience in maintaining radar contact by manipulating the servos during an approach. During the trainee's initial encounters with the system, the servos can be frozen and the total area will be used to control target size. The display area is defined specifically in terms of the slopes from the maximum and minimum total display points at the one-mile mark to the radar, or from the top and bottom of the one-mile hashmark plus some leeway to the radar. Three values are ultimately obtained: (1) the halfsize of a normal radar target, (2) the size from the midpoint of the target to the top of the display area, (3) the size from the midpoint of the target to the bottom of the display area. The minimum of these three values is added to the midpoint of the target on the other display area to obtain the upper screen coordinate of the target, and subtracted from the midpoint for the lower screen coordinate. A final check is made to confine the target properly within the display area. Ultimately, five values are returned to RADAR for transmission to CPU 2 for display: (1) the range of the target as an X-value in the Megatek display area, (2) the top of the azimuth target as a Y-value, (3) the bottom of the azimuth target as a Y-value, (4) the top of the elevation target as a Y-value, (5) the bottom of the elevation target as a Y-value.

DISPLAY SIMULATION

Any controller training system must have a realistic display to give the trainee firsthand experience in conducting an approach. At the same time, flexibility in the display is desirable to improve training effectiveness. The GCA-CTS uses a commercial graphics display to simulate the PAR radar display and at the same time provide the degree of flexibility needed in a training system. The major role of the Megatek display is to simulate the radar used at a controller installation. It can also be used, however, to enhance the training process by providing track histories and text for emphasis of particular events in an approach. The GCA-CTS PAR display is shown in Figure 51. The block diagram for the display software is in Figure 52.

The GCA-CTS display routines reside in foreground on CPU 2. The display routines act as the interface between the GCA-CTS software and the Megatek graphics library. The routines in the graphics library create the display list, a series of instructions acted upon by the Megatek hardware to draw and move lines and characters on the display. The GCA-CTS display routines call the library routines necessary to draw the precision approach radar (PAR) display, and to move the targets and trails realistically down the glideslope and centerline. The following pictures are presently available to be displayed on the Megatek:

a. Azimuth display

b. Azimuth hashmarks

c. Elevation display

d. Elevation hashmarks

e. Azimuth target

f. Azimuth trail

g. Elevation target

h. Elevation trail

i. Wind chart

j. Character string

k. Long azimuth trail

l. Long elevation trail

m. Touchdown reflector on elevation

n. Touchdown and centerline reflectors on azimuth

The reason for separating pictures in this way is to allow flexibility in displaying the portions of the picture the trainee should pay special attention to. In addition, the string of characters may be used to alert the

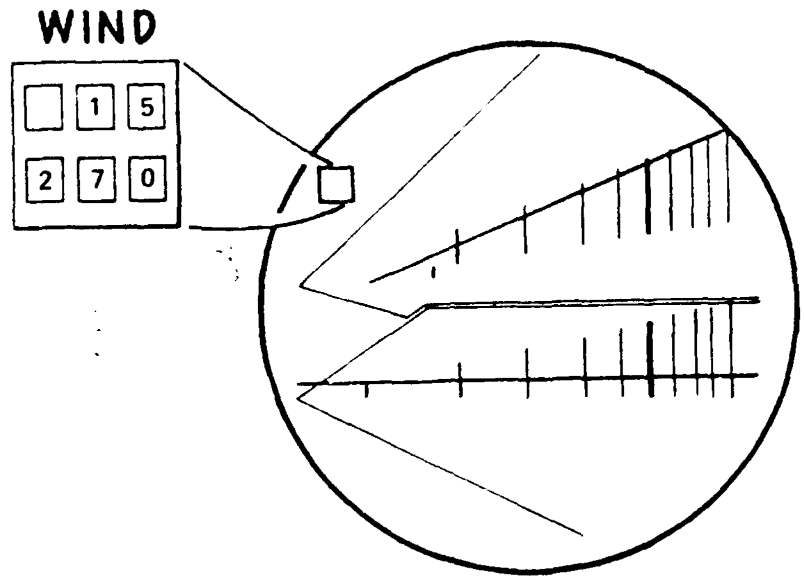


Figure 51. GCA-CTS PAR Display

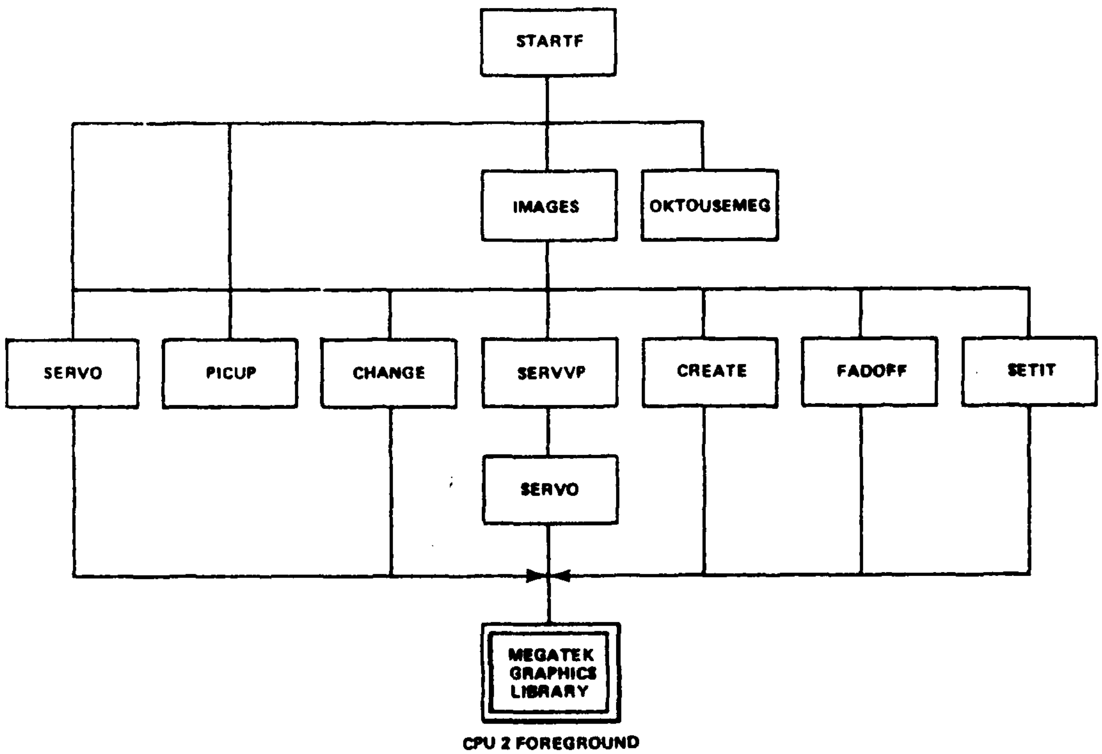


Figure 52. Display Block Diagram

trainee to information being presented on the CRT, or to point out some special condition on the display. It is generally not present, however, to maintain the realism of the PAR display.

The Megatek routines in the GCA-CTS graphics library are not re-entrant; that is, in a tasking environment, a routine cannot be interrupted to let another Megatek routine operate without creating havoc with the Megatek display list. Two different methods were implemented to deal with this restriction. In general routines were designed to be called from a central location, the routine called IMAGES. All routines were designed to be function-specific to reduce the chances of two routines being called at the same time from different tasks. Ordinarily this would have been sufficient, but it was considered desirable to have the servo continue to operate between runs and when a demonstration was in progress. This is true because it provides a way to determine if the trainee is playing with the servo when he shouldn't be. To do this, a task was created, called SERVUP, to obtain the position of the servo. This task is protected, as are all the display routines called by the foreground executive STARTF, by a function called OKTOUSEMEG, which returns a logical true, locks out other display routines, and allows the caller to update the display list.

INITIALIZATION. There is a sequence of calls to routines in IMAGES to prepare the Megatek for display. These routines create display lists, turn on the joystick or servo, move the hashmarks to the centered position on the course and glideslope, task SERVUP and start the display processor.

TARGET MOVEMENT. An incoming aircraft is displayed in two views, top and side. As the aircraft approaches the runway it shrinks. It also appears to move faster as the range to touchdown grows shorter to conform to the logarithmic nature of the display. In addition, a trail of phosphor glow is perceptible and is used by the final controller to determine the past movement of the aircraft. This trail must be simulated in the display.

One way to accomplish this target and trail is to insert the latest update each half second. (The half second corresponds to the PAR sweep, which is not being implemented here.) This method would be too space consuming. An alternate method might be to use absolute vectors and redraw them all each half second. This method is very time consuming. To avoid both these problems, the part of the display list which draws target and trails is treated as a circular buffer. The latest update is written over the least recent update. This keeps the target and trails a fixed size while minimizing the time needed to update the picture. While the picture is being updated, a track history is being saved. These pictures are created by inserting every fifth vector into the display list while the pictures are off. At the end of the run, the pictures are turned on. In this way the optimal goal of combining a realistic display with a meaningful training display is achieved.

RANGE AND TIME-RELATED SUBROUTINE SCHEDULING

There are many actions which the trainee is supposed to perform within a certain time after another event or at a certain range from touchdown. It seems most natural to have separate routines check for omissions of these actions. There are two obvious ways to handle these "omission check" routines given the software environment of the system: make each of these a task, or schedule them via the FORTRAN .QTSK package. For the following reasons, neither of these solutions was chosen.

The problem with making these routines tasks is that, during a run, several of these tasks might be active at one time and each would require stack space. This would be an unacceptable situation because the system would quickly use up its available stacks.

The Fortran .QTSK package could not be used for scheduling for two reasons. First, the instructor is allowed to freeze phase 2 and 3 problems and there is no convenient way of temporarily disabling the Fortran package. Second, performance measurement was designed to operate either in real time (in phase 2), or non-real time (after a phase 3 problem) and PMS often relies on information from these omission check routines. Unfortunately, the Fortran package is designed so that the user cannot bypass the real-time clock and maintain scheduling.

For these reasons, a set of scheduling and calling routines was designed. The set can be divided into two groups: one for time scheduling and the other for range scheduling. We will first discuss what these two groups do similarly and then we will mention their differences.

SCHEDULING AND CALLING. The scheduling routines (RNGSCHD and TIMSCHD) maintain linked lists in which each element contains the entry point of a routine and a pointer to the next element in the list. The elements are ordered such that the routine at each link is to be called after the routine of the previous link and there is a pointer to the first element. Whenever RNGSCHD or TIMSCHD is called, it first looks to see if there is more room in the queue. If so, it looks for the spot in the list where this routine belongs and inserts it.

The calling routines (RNGCAL and TIMCAL) each call any routine at the top of their respective lists whose time or range has come. Both TIMCAL and RNGCAL have half-second periods (defined by a user clock during phase 2 and by times retrieved form the activity file after a phase 3 run). RNGCAL and TIMCAL then return freed elements to the queue for reuse by the schedules.

RANGE SCHEDULING. The actions which are to be done by RNGCAL at a given range have been divided into two groups: call a routine or place a particular SUS phrase number into a word in common. In order for the model controller to tell if a particular phrase is acceptable, RNGSCHD was designed to place a phrase number (passed to it as an argument) into common words which can be accessed and/or changed by other routines.

Because range-related actions often come in pairs (beginning and end points) and because the ranges are usually known at the start of the run, a routine (PCHK) was designed to facilitate range scheduling. It is called with a beginning point, an end point, an address in common, a phrase number, and a routine name. It will then schedule the appropriate actions.

TIME SCHEDULING. TIMSCHD, in addition to the characteristics it shares with RNGSCHD, was designed to allow a routine to be deleted from the time list. This means that the scheduled routines can assume that, if they get called, an omission has occurred and the appropriate performance measurement action can be undertaken. Therefore when the trainee has accomplished a time-related action, TIMSCHD is often called to delete that action's omission check routine from the list.

CONTROLLER MODELS

A ground controlled approach is actually a collaboration of three different controllers. In designing the GCA training system it was recognized that all three should be simulated in order to effectively train and test the controller trainee. The three controllers are perceived by this system as follows:

a. Pattern Controller. The pattern controller feeds aircraft to the final controller. He issues waveoff instructions to the pilot and, if necessary, provides a turn called a dogleg, designed to put the aircraft on a course which will intersect the extended runway centerline at an angle of 20 to 40 degrees. Should the final controller not respond as expected, the pattern controller will wave the pilot off.

b. Tower. The tower issues and cancels clearance. All communication to and from the tower is through the GCA student panel.

c. Final Controller. The final controller is responsible for communication with the incoming aircraft from the point at which he reports radar contact until the aircraft has landed, or until the pattern controller has accepted the handoff for a waveoff or normal termination of a low approach or touch-and-go. His transmissions include glidepath and course position and trend messages, range-related advisories, emergency procedures such as waveoffs or gyro failure instructions and clearance information.

Block diagrams of the controller simulations are drawn in Figures 53, 54, 55, and 56. The three controller simulations provide three services to the GCA.

a. Realism. The controller simulations approximate the conditions in a radar station, including human error. For example, the pattern controller may fail to release the frequency of the aircraft to the final controller during the handoff. This should make the trainee's transaction to a genuine station smooth.

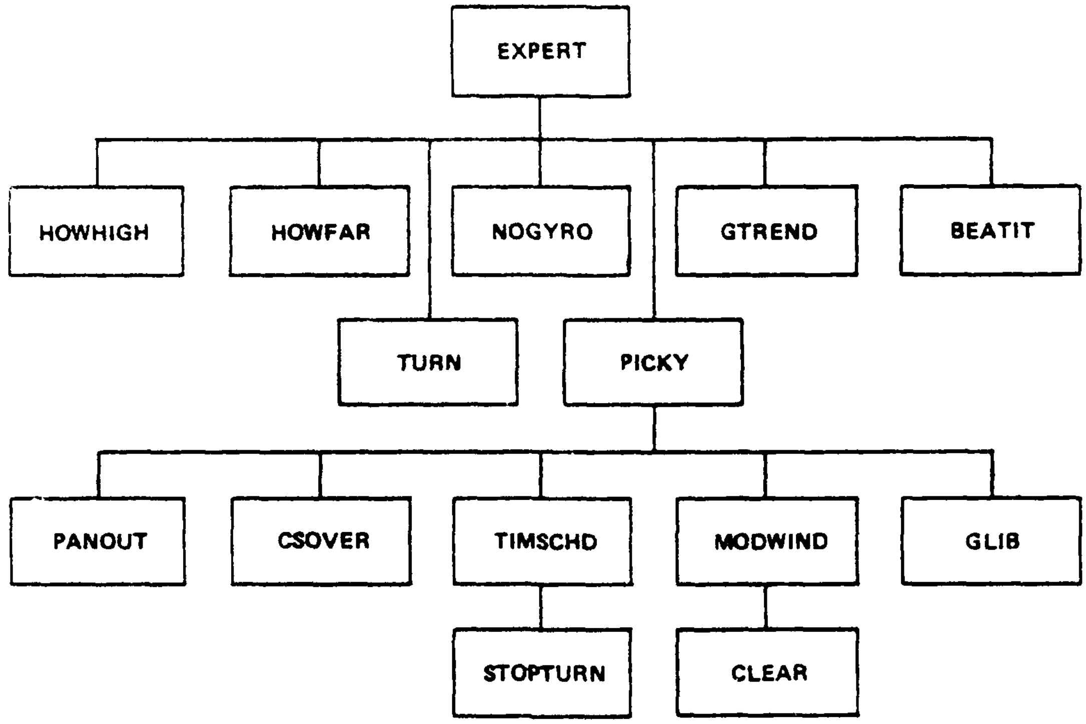


Figure 53. Model Controller Executive Block Diagram

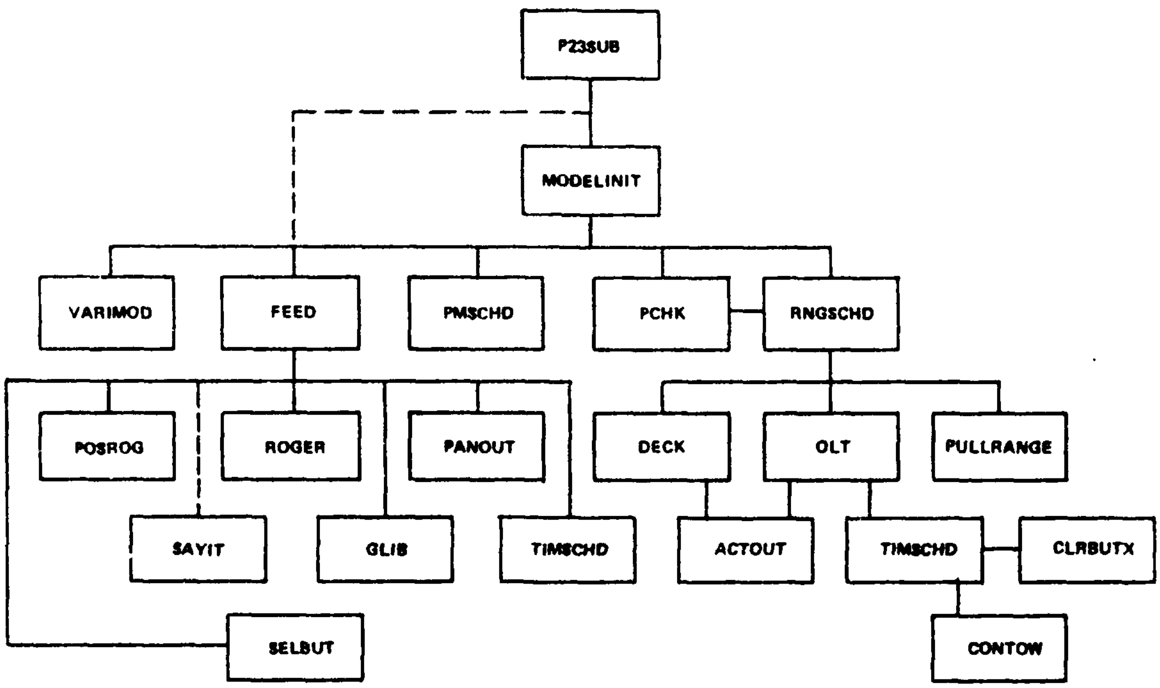


Figure 54. Model Controller Initialization Block Diagram

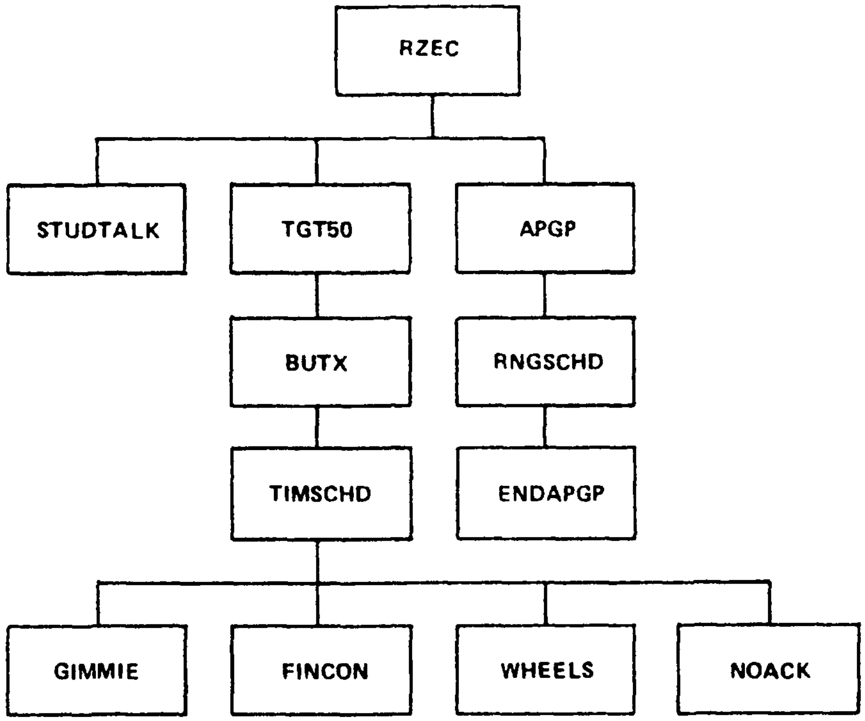


Figure 55. Model Controller Run Related Routines

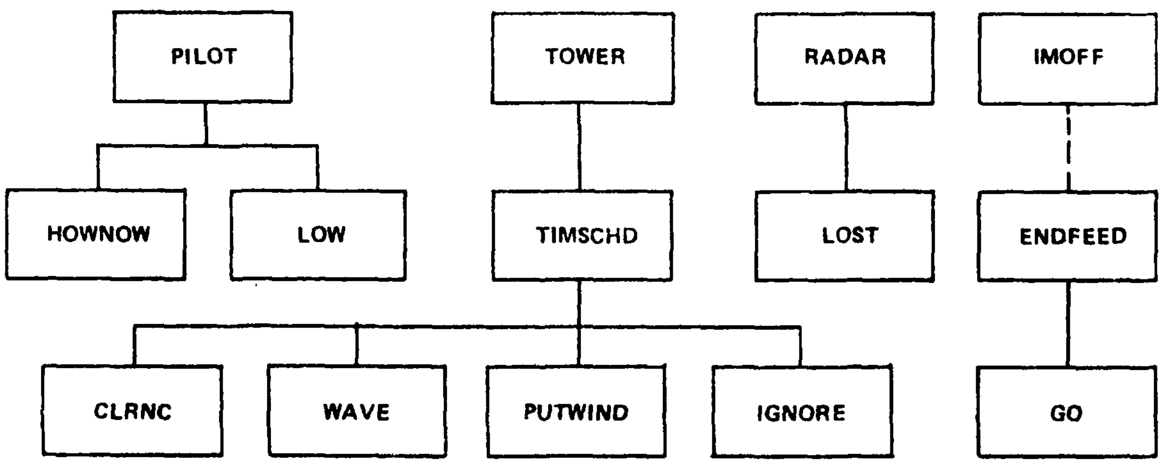


Figure 56. Model Controller Block Diagram for Tower, Pilot and Radar Related Routines

b. Demonstrations. The controller simulations are capable of interacting together with the final controller simulator emitting audible best-choice advisories. In this way the trainee becomes familiar with ideal performance.

c. Speech Recognition. Because the controller simulations determine at different points in a run which transmissions are most correct, speech recognition can be improved by examination of the best choice transmissions. For example, "Turn right heading" and "Turn left heading" may be confused by speech recognition. The controller model, however, knows which turn is appropriate. Speech recognition can obtain the proper turn from the controller model and use this information to understand the trainee's phrase if there are two possible recognition choices. While this may still contribute to misrecognition, as when the trainee gives the turn in the incorrect direction, and the recognition is low confidence, it was decided to give the trainee the benefit of the doubt, when possible, to avoid trainee frustration.

In designing the controller simulations it was decided to keep the three controllers separate groups of routines interacting together. In terms of performance, however, the model controller can be perceived as three major areas of communication, the instigator of each communication being the first simulation mentioned of the pair.

The first communication area is between the pattern controller and the final controller during the handoff.

The second communication area is that of the final controller to the pilot during an approach.

The third communication is conducted between the final controller and the pattern controller during a waveoff, or upon completion of a touch-and-go or low approach.

PATTERN TO FINAL CONTROLLER. This area begins with the controller's instructions to the pilot when the aircraft is not visible on the PAR display and terminates when the first final controller-pilot communication occurs. The transmissions included in this area are controlled in the routine FEED and encompass the following: (1) Transmission of missed approach procedures to the pilot, (2) transmission of the dogleg turn designed to put the aircraft on course, if necessary, and (3) transmission of the handoff to the final controller. This transmission includes the aircraft type, call sign, present position (right/left base, straight-in) type of approach, no-gyro advisory, if any, and frequency.

FINAL CONTROLLER TO PILOT. This area begins when the pattern controller has released the aircraft frequency and terminates in one of four mutually exclusive ways.

It will terminate if the pilot waves off for any reason. It will terminate at decision height if the aircraft is flying a low approach. If the aircraft is flying a touch-and-go, it will terminate over landing threshold.

during a normal full-stop approach, it will terminate after the final controller releases the frequency. The series of communications involved in this area include the following: (1) the initial contact between the final controller and the pilot, (2) the turn to final, (3) the course/glidepath position/trend and range transmissions, (4) emergencies, such as lost radar contact, no-gyro instructions and waveoffs, (5) clearance exchange between the final controller and the tower. The clearance exchange is handled through the GCA-CTS trainee panel.

The final controller must give a smooth flow of advisories with no more than a five-second delay between advisories. At the same time the final controller must keep the air clear whenever possible to allow the pilot to speak.

FINAL-PATTERN CONTROLLER. This communication exchange occurs only if the pilot fails to execute a normal full-stop approach. If the plane is executing a low approach or touch-and-go, or if the pilot waves off for any reason, this third area of controller communication is executed. The exchange terminates when the pattern controller has accepted the aircraft from the final controller. The exchange includes the final controller giving the handoff to the pattern controller and the pattern controller's acceptance of this handoff.

MODES OF OPERATION. The controller model performs in two modes. When a demonstration is in progress, the final controller routine PICKY selects the most appropriate message from the array of acceptable messages located in controller common, builds a complete message including call sign if needed and sends it to the subroutine GLIB. GLIB prepares this message for one of the speech output devices.

The three controller communication areas had to be kept mutually exclusive to fit in the overlay areas available to them. Therefore, when the pattern controller simulations FEED and ENDFEED are active, special routines are called to keep the final controller, EXPERT, separate. These routines are only called when a demonstration is in progress and access the speech output devices directly through GLIB. All other final controller routines fill the common area reserved for controller advisories and which are accessed by PICKY. The message chosen by PICKY is based in part on the following hierarchy: (1) emergencies, including at decision height if a waveoff is required, (2) clearance, no-gyro instructions, (3) range-related advisories, (4) glidepath position and trend messages, (5) course position messages, (6) turns. There is special handling for no-gyro turns during demonstrations. If PICKY selects a no-gyro turn, it schedules STOPTURN to be called when the aircraft has reached the point determined by the TURN routine. Until the turn has been stopped, no other advisories are given. To maintain the five second rule, several shorter no-gyro turns may need to be given.

If the trainee is executing an approach, the controller common area is used as mentioned before for clarification of trainee responses for speech recognition. On occasion, the pattern controller simulators FEED or ENDFEED require a specific response within a given period of time. When this is true, FEED schedules a routine called HOLD to be called when it is no longer possible to give an acceptable response. HOLD informs FEED that the trainee

has not responded correctly by transmitting (using XMT, a Fortran library routine) a message to a mailbox in common. FEED also activates a listening task called SAYIT to notify FEED if a correct transmission is made. Then FEED waits on a REC (the counterpart to XMT) for whatever message comes in first. Whichever routine transmits its message first, SAYIT or HOLD, that routine cancels the other routine. In this way a second message transmission should never get in.

MESSAGE FILL. When the pattern controller simulators are not active, many routines are called to fill the controller common with a list of acceptable messages for any point in a run. Some of these routines, specifically the range-related routines, are scheduled during controller initialization. Other routines are dependent on specific occurrences during a run. An example of this may be lost radar contact. These events can be more readily discerned by other modules in the GCA-CTS system, and these modules will call the appropriate routine to fill controller common. In the above example, the routine RADAR would call LOST to put the emergency message into controller common. Other events are caused by modules who alert the proper final controller routines. Thus if the aircraft/pilot/environment simulation (APE) conducts a waveoff, the final controller to pattern controller exchange will be initiated. Other routines which are time-related are scheduled by final controller routines. For example, "begin descent" should be given between 10 and 30 seconds of "approaching glidepath". There are other restrictions on this particular advisory, but the initial placement of the message in common is scheduled as described.

MESSAGE REMOVAL. When the pattern controller is not active, a routine named STUDTALK is called by FZEC for the final controller executive, EXPERT. This routine accepts the latest advisory received from the trainee by speech recognition and removes it from the list of acceptable messages. It is also necessary to remove messages which must be given within a specified time period. If an advisory must be given at a specific time, the starting wait time and legal maximum in relative number of seconds are stored in common. The index of the message to be removed is also stored. If the maximum wait time has been exceeded, the appropriate advisory is removed from the list of acceptable messages. There are two other routines which remove advisories from the list of acceptable messages. ENDAPGP removes the "approach glidepath" advisory and updates the activity file buffer. PULLRANGE removes the mile-mark advisories. Thus there is a continuous revision performed on the list of acceptable advisories used by the controller simulations.

MODEL CONTROLLER TURN ALGORITHM. Before giving the turn algorithm, let us first define some terms which appear in the algorithms.

a. A distinguishable heading is any compass heading from 0 to 359, inclusive, which is congruent to 0, 2, or 3 modulo 5.

b. Final Turn Heading (FTH).

$$\text{Let } H_A = 160 - \sin^{-1}\left[\frac{S_w \sin(160-H_w)}{S_A}\right]$$

where S_W = mean windspeed (knots)

S_A = aircraft "final approach airspeed" (knots)

H_W = mean wind direction (degrees)

and $\sin^{-1}$ is the arcsin function for "degree arguments,"

then the "Final Turn Heading" is

(1) H_A, if H_A is a distinguishable heading, or else

(2) the nearest lesser/greater distinguishable heading from H_A, if the current wind is (a) a pure headwind or left crosswind, or (b) a right crosswind.

The final turn heading is a distinguishable heading which closely approximates that heading H_A which would, in a constant wind of S_W knots from H_W degrees, produce an aircraft ground track parallel to the simulated runway (Runway 16).

Representative values of FTH as a function of aircraft speed and wind state are found in Table 26.

c. Turn zones. The aircraft is said to be in "Turn Zone"

"0" if the course centerline interacts the displayed target's "middle" third, or ...

"1" if the course centerline intersects the displayed target's "upper" or "lower" thirds, or ...

"2" if the course center line does not intersect the displayed target, but the endpoint of the target nearest the course centerline lies within one-half target width of the course centerline, or ...

"3" otherwise.

The algorithm performs the following sequence of steps to turn the aircraft to the final approach course:

0. Compute (or estimate) FTH based on estimated aircraft speed and estimated current mean windspeed and heading. Go to step 1.

TABLE 26. REPRESENTATIVE VALUES OF THE FINAL TURN HEADING

Aircraft Final Approach Airspeed	Mean Wind Heading	Final Turn Heading (FTH) Mean Wind Speed (Knots) 0	10	20	30
98	70	160	153	147	142
	130	160	157	153	150
	150	160	158	157	155
	160	160	160	160	160
	170	160	162	163	165
	190	160	163	167	170
	250	160	167	173	178
115	70	160	153	148	143
	130	160	157	155	152
	150	160	158	158	157
	160	160	160	160	160
	170	160	162	162	163
	190	160	163	165	168
	250	160	167	172	177
130	70	160	155	150	145
	130	160	157	155	153
	150	160	158	158	157
	160	160	160	160	160
	170	160	162	162	163
	190	160	163	165	167
	250	160	165	170	175

TABLE 26. REPRESENTATIVE VALUES OF THE FINAL TURN HEADING (CONT)

Aircraft Final Approach Airspeed	Mean Wind Heading	Final Turn Heading (FTH) Mean Wind Speed (Knots) 0	10	20	30
156	70	160	155	152	148
	130	160	158	155	153
	150	160	158	158	158
	160	160	160	160	160
	170	160	162	162	162
	190	160	162	165	167
	250	160	165	168	172

1. If the aircraft has, at any time thus far during the approach, been given a turn to the FTH, exit; else go to step 2.

2. If the aircraft is on or to the left of/to the right of course, and the aircraft has, since the last turn advisory was issued (or if none has yet been issued), moved into Turn Zone:

   3; turn the aircraft to a heading of FTH $\pm 15°$ rounded to nearest 5°

   2; turn the aircraft to a heading of FTH $\pm 10°$ rounded to nearest 5°

   1; turn the aircraft to a heading of FTH $\pm 5°$ rounded to nearest 5°

   0; turn the aircraft to a heading of FTH rounded to nearest 5°

   Go to step 1.

After the turn to final is complete, a second turn algorithm is used to provide corrective and counter-corrective turns in a manner similar to the technique which is taught to the trainee.

AUTOMATED VOICE

There are two devices in this system for producing human voice or its simulation. One system, digitized voice, records and plays back audio input. The other system, VOTRAX, generates a simulated voice. The advantage of the digitized voice is that it sounds more realistic, and the advantage of VOTRAX is that it uses less disk space.

THE VOICE EXECUTIVE. All requests for verbal output (except during REPLAY) are marshalled by a set of four routines shown in Figure 57. The major purpose of these routines is to:

a. Rearrange the input list, if necessary, to conform to a particular format,

b. Determine the unit to be used,

c. Schedule the output, and

d. Activate the output device.

There is a circular queue (PRQUE) which is filled by GLBF and read by DONE. When GLBF is finished putting its arguments into PRQUE, it tasks DONE if DONE is not presently active.

DONE merely goes through PRQUE and sends the contents to the correct output device. When DONE is through passing arguments to the output device it checks to see if this is the demo mode. If so and if the pattern controller has handed off, DONE sets a flag (CTCALEXP) which indicates that EXPERT is expected to provide further verbal output.

If the output device is VOTRAX and the mode is a phase 3 or P-run, DONE goes through PRQUE again and fills an array with records to be placed in RPLACT. It then calls ACTOUT with this array.

DIGITIZED VOICE. This is a set of routines that records and plays back voice data using the speech digitizer (device 31). These routines can be used for producing speech for a number of purposes. The uses include production of "canned" phrases for prompting the student during phase 1, playback of selected phases, and playback of a digitized voice file during replay.

Due to the fact that this data channel device requires two 2K buffers, it was deemed advisable to put these buffers in extended memory. Therefore, after setting up the data channel map, the buffers are mapped into extended memory and free pages are mapped into user address space. This bookkeeping is done by SYSINIT and SPNIT and must be accomplished before the device is used.

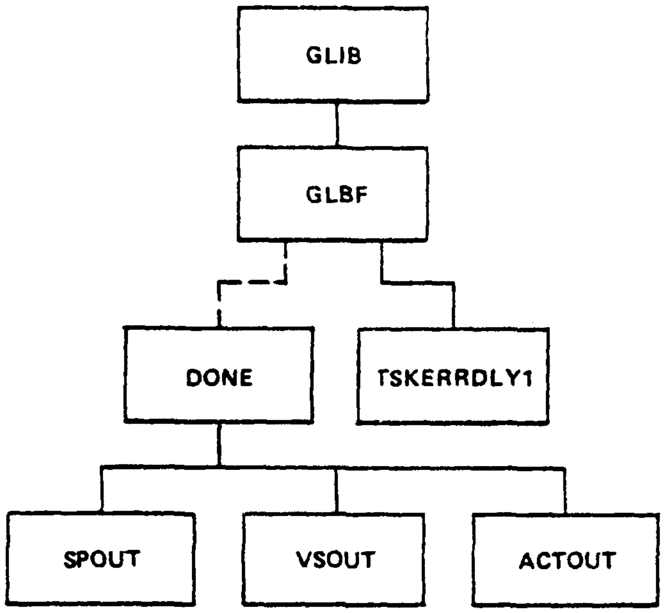


Figure 57. The Structure of the Voice Executive

Recording. Data recorded by the speech digitizer must be moved from one of the buffers in extended memory to a file on the disk. The digitizer can record in either of two modes: over the last item entered, or in the next available space. The first mode is used during phase 1 when the student is being taught the phrase list. He is asked to repeat each item until it sounds correct. It is assumed that the most recent version is the best. Therefore, each time he repeats a phrase it is recorded and the previous repetition is erased.

The second recording mode is used in phase 3 when a real-time recording is desired. In this mode, the buffered data are placed in a new spot on the disk. The choice of recording mode is controlled by an argument passed to SPIN.

Playback. Digitized playback consists of moving data from the disk to one of the buffers (handled by SPBUF) and activating device 31 in the play mode (done by STRTPLY). Playback, like recording, is in either of two modes: playback of a particular phrase, or playback of the replay file. The first of these is used in phase 1 and the second during replay.

In non-replay modes, SPOUT, after buffering up data, will start the device playing by calling STRTPLY. However, during replay it is necessary that the replaying of the three files (student activity, radar display, and

digitized voice) begin as close to simultaneously as possible. Therefore, in this mode, SPOUT gets the buffers filled with data from the digitized speech file, but does not start the device playing. Then, when the other replay components are ready to start, REPLAY calls STRTPLY to start the device.

Playback/Record. The speech digitizer was designed to allow it to "record itself." That is, if prerecorded phrases are used during phase 3, it is necessary that those phrases be recorded in the digitized voice file so that they will be output along with the rest of that file during replay. However, the device cannot record and play simultaneously. Therefore, the interrupt service routine which handles device 31 (SPIS) determines whether the system is in record, play, or play/record mode. If it is in the play/record mode then, after a buffer is played by SPOUT, the same buffer is recorded on the disk by SPDMP.

VOTRAX. The voice generation unit (VGU) is controlled by three routines whose purpose is to decode output requests, read the required phonemes from the phoneme file, and output the phonemes to the VGU.

The routines must do a small amount of decoding on the input arguments before the arguments can be used a phrase numbers. One such argument (VXKIL) is interpreted as an order to kill the spooling to the device. Another argument (VXDIG) means that the next argument is to be taken as a digit to be converted into the appropriate phrase numbers. If any argument is not within the range of legal phrase numbers, a message is sent to the bug file.

The design of the routines had to address the fact that it takes a certain amount of time to read the phonemes in from the disk and that this could cause unwanted pauses in utterances. Therefore, the routines maintain three buffers of phonemes to be output. The routine which sends the phonemes to the VGU (WRFRAZ) first tasks another routine (RDFRAZ) to fill the buffers from the phoneme file. WRFRAZ waits until RDFRAZ gets at least two buffers ahead (or finishes) before activating the VGU.

PERFORMANCE MEASUREMENT

Performance measurement is designed to evaluate a trainee's performance on specifically defined tasks. Description of tasks, and the methods used for evaluation, follow.

PERFORMANCE MEASUREMENT VARIABLES. Performance data are required to provide the following capabilities:

a. Real-time error detection and freeze in phase 2.

b. Adaptive problem selection.

c. Student feedback in the form of annotated replay and performance summaries.

d. Instructor feedback emphasizing overall progress.

e. Performance test scoring after misrecognition correction.

When an error is detected, a bit is set in the appropriate performance measurement variable (PMV) listed in Tables 27-46. The bits are then used in scoring and feedback. For some purposes, such as for freezing in phase 2, only an error indicator is needed. Adaptive problem selection and feedback on phase 3, on the other hand, require a quantitative measure of performance on specific tasks in which errors are weighted by their relative importance. In the PMV descriptions in the tables which follow, both the error bits and the contributions to the particular PMV score are defined. All PMV scores are in the range of $0 < s < 100$.

In general, the PMVs are arranged so that all the bits concerning a particular topic are together in a word. The exception to this is PV00 which contains things which must be done once during each run. The reason for this is that all of these (and only these) bits must be set at the beginning of a run, and it was thought to be better not to have such bits scattered through various PMVs.

Consequently, the scoring columns are omitted from the table on PV00. Instead each of these bits is discussed in the PMV which contains related bits. The table on PV00 indicates which word to refer to for scoring information. In PV01 through PV19 the column headed "Bit of PV.." is used to indicate which bit carries the information. If an entry has an asterisk after it, this indicates that the bit resides in PV00 and the number after the comma specifies the corresponding number in the present PMV.

Occasionally, two numbers in the "Bit of PV.." column are followed by a brace. This indicates that the bits represent events which are mutually exclusive. If either of these bits are set, the points indicated are to be deducted. The two bits are maintained for feedback purposes.

Some PMVs, such as PV07, have two extra columns headed "Error" and "Counter." These are used when an action would typically occur several times during a run, such as the issuance of trend calls. The numbers in these columns indicate words of the PV.. array which are used as counters of events. The word indicated in the "Counter" column contains the number of times that an action should have been taken and the word indicated in the "Error" column records how many times the action was not taken correctly. That is, the "Error" word divided by the "Counter" word is the percentage of times that the action was not properly taken. The description in the "Controller Action" column describes the action which the bit is intended to record if it is not set.

PERFORMANCE MEASUREMENT ROUTINES. Performance measurement is an integral portion of the training process. The problem arises on how to score all the pertinent performance areas. This is resolved by breaking down the scorable components of a run into independent performance variables (PVs), as mentioned previously. Before a run, variables are flagged for later scoring as specified by the training file. Figures 58 and 59 are block diagrams for PMS. The initialization routine PMINT calls individual initialization routines for each performance variable to be scored. The PMS initialization routines fill tables with entry points to routines which investigate the events relevant to the PVs to be scored. During PMS execution, an event triggers a table lookup.

TABLE 27. PV00, ACTIONS DONE ONCE EVERY RUN

Controller Action	Bit of PV00	PMV with Related Bits
Pattern Controller monitored	0	1
"Approaching Glidepath" said	1	4
"Do Not Acknowledge..." said	2	4
"Begin Descent" said	3	4
"At Decision Height" said	4	9
Clearance requested	5	10
Clearance/wind or waveoff given	6	10
"Over landing threshold" given	7	11
Rollout instructions given	8	12
Handoff to pattern controller done	9	12
Frequency released	10	12
No-gyro announced	11	13
"Make half..." not yet said	12	13

TABLE 28. PV01, ACCEPT HANDOFF COMPOSITE

Controller Action	Bit of PV01	Partial Credit
A. Monitor feeder controller ICS	0*,1	10
B. Monitor proper frequency as specified in the handoff	2	10
C. Acknowledge handoff		
1) Acknowledgement given prior to radar contact	3	10
2) Acknowledgement given within 10 seconds	4	10
D. Report radar contact		
1) Radar contact reported prior to radio check	5	10
2) 50% of target on display at report	6	15
3) Report not later than 10 seconds after 50% target appearance	7	15
4) Call sign correct	8	5
5) Radio frequency correct	9	5
E. ICS off, radio frequency selected		
1) Pattern controller does not relinquish frequency, "Give me..." request made within 15 seconds	10	5
2) Pattern controller relinquishes frequency and "Give me..." not used	11	
3) When pattern relinquishes frequency, ICS is deselected	12	5

*Resides in PV00

TABLE 29. PV02, RADIO CHECK COMPOSITE

Controller Action	Bit of PV02	Partial Credit
A. Radio Contact		
1) Within 30 seconds of 50% target appearance	1	10
2) Proper frequency selected	2	10
3) Mike keyed	3	10
4) Call sign used	4	10
5) One of the following given:	5	10
a) "How do you hear..."		
b) "Wheels..."		
c) "Turn...heading"		
d) "Turn..."		
6) Mike unkeyed within 3 seconds and left unkeyed 5 seconds	6	20
B. Speech quality		
1) Pilot responds "Loud and clear," or	7	30
2) If pilot responds "Weak...,"		
a) Student answers "how...now," unkeys within 3 seconds and leaves unkeyed 5 seconds	8	15
b) Pilot can respond "Loud...," i.e., V.U. level normal	9	15

TABLE 30. PV03, TURN-TO-FINAL COMPOSITE

Controller Action	Bit of PV03	Partial Credit Turn	Partial Credit Straight-In
A. Accuracy of turn vectors, if given. (Score is given a weight of .6, score for B weighted .4; for a straight-in approach, the entire 100 points is given on B 1 and 2)			
1) Turn in proper direction	7,10,13	40	
2) Call sign correct	9,12,15	20	
B. Quality of turn or initial control			
1) At 6 miles (3 for short approach) target is within 2 target widths of cursor	1	10	30
2) At 5 miles (2 short approach) target intercepts azimuth cursor in target zone 1 or 2	2	20	70
3) More than 1 turn used to turn aircraft onto final	3	10	

TABLE 31. PV04, APPROACHING GLIDEPATH COMPOSITE

Controller Action	Bit of PV04	Partial Credit
A. Approaching glidepath		
1) Transmission given	1*,0	10
2) Call sign and "over" needed and used	1	5
Call sign and "over" not needed and not used	2	
3) Transmission given when aircraft is within the correct range	3	5
4) Transmission given only once during final approach	4	5
B. Do not acknowledge		
1) Transmission given only once	2*,5	10
2) Correct call sign used	6	5
3) The phrase is not followed by "over"	7	5
4) Transmitted prior to "begin descent"	8	5
C. Begin descent		
1) Transmission given	3*,9	10
2) Transmitted within 10-30 seconds after "approaching glidepath"	10	5
3) Glidepath cursor intersects upper 1/3 of target when advisory given	11	10
4) Transmitted only once during the approach	12	5

Range table for A.3:

Aircraft Speed	Acceptable Range (Miles)
90	0.25-0.75
120	0.33-1.00
140	0.38-1.16
160	0.44-1.33
200	0.55-1.67

TABLE 31. PV04, APPROACHING GLIDEPATH COMPOSITE (CONT)

Controller Action	Bit of PV04	Partial Credit
D. Wheel check		
1) Transmission given prior to "approaching glidepath" when pilot has not said "wheels down"	13	15
Transmission not given after pilot has said "wheels down"	14	
2) Correct call sign and "over" used	15	5

*Resides in PV00

TABLE 32. PV05, HEADING VECTORS COMPOSITE

Controller Action	Bit of PV05(0)	Array Element of Error	PV05 Counter	Weighting Factor Applied to Percentage Error
A. While range greater than 5 miles; all turns evenly divisible by 5°	1	1	9	.1
B. Turns must not be of 1°	2	2	10	.1
C. All heading vectors				
1) Direction of the turn and heading digits correspond such that the direction advised causes the smaller turn	3	3	11	.2
2) A counter-corrective turn made within 8 seconds when a turn of more than 120° is given	4	4	12	.05
3) Target enters zone 3 from zone 2, a heading correction given within 20 seconds. This check is initiated when target has been in zones 1 or 2 for 1/2 mile, or at 5 miles (2 for short approach), whichever comes first	5	5	13	.15
The heading given in the "Heading..." message the same as previously assigned	7	7	14	.25
"Heading..." not used more than 5 times in an approach	8	8	14	.15

TABLE 33. PV06, AZIMUTH POSITION AND TREND COMPOSITE

Controller Action	Bit of PV06(0)	Array Element of PV06 Error Counter		Weighting Factor Applied to Percentage Error
A. Position calls				
1) Position call correct	1	1	5	.5
2) "Well" followed by a corrective turn within 3 seconds, or "correcting"	2, 3	2	6	.25
B. Trend calls				
"Correcting" used only when target is closing with centerline	4	4	7	.25

TABLE 34. PV07, GLIDEPATH POSITION AND TREND COMPOSITE

Controller Action	Bit of PV07(0)	Array of Error	Element PV07 Counter	Weighting Factor Applied to Percentage Error
A. For all glidepath messages, "begin descent" has been given	1	1	8	.10
B. Position calls				
1) Position correct	2	2	9	.15
2) A position call made whenever target changes zones, unless superseded by a priority call	3	3	10	.15
C. Trend Calls				
1) Trend correct	4	4	11	.15
2) Trend issued if the target moves from one zone to another	5	5	11	.15
3) Trends not issued successively except in well zone	6	6	11	.15
4) Trends do not separate identical position messages except in well zone	7	7	11	.15

TABLE 35. PV08, RANGE CALL COMPOSITE

Controller Action	Bit of PV08(0)	Array of Error	Element PV08 Counter	Weighting Factor Applied to Percentage Error
A. All range calls made once the first one is made or 5 miles is reached, whichever comes first, unless superseded	11	1	5	.6
B. The call made within ±0.1 mile of the mark	12	2		.2
C. Correct range used	13	3		.2

TABLE 36. PV09, DECISION HEIGHT COMPOSITE

Controller Action	Bit of PV09	Partial Credit
A. Decision height call		
1) Call given	4*,1**	25
2) Target not touching cursors and call was followed by highest priority correct position (bits 3-6 indicate what the highest priority call was)	2	25
B. Range		
1) DH announced within .80 miles from touchdown	7**	20
2) DH announced prior to .7 miles from touchdown	8**	25
C. Call is made only once during the approach	9	5

* Resides in PV00
**Safety error

TABLE 37. PV10, CLEARANCE COMPOSITE

Controller Action	Bit of PV10	Partial Credit
A. Clearance requested	5*,0	
1) Initial clearance request made after 3.1 miles	1	10
2) Initial clearance request made prior to or at 2.9 miles	2	30
3) Clearance not received and second request posted between 2.1 and 1.9 miles, or,	3	10
Clearance received and not requested again	4	
B. Issuance of clearance when received from tower	6*,13	
1) Correct wind information given	5	10
2) Wind issued after clearance is received from tower	6	10
3) Clearance issued after received from tower	7**	5
4) Clearance issued after wind advisory	8	5
5) Clearance issued prior to 1 mile	9	20
or		
C. Clearance problems leading to a waveoff		
1) If clearance is not received		
a) Reason and waveoff issued prior to 1.3 miles, option not given	10**	35
b) Proper missed approach transmission used	11	15
or		

TABLE 37. PV10, CLEARANCE COMPOSITE (CONT)

Controller Action	Bit of PV10	Partial Credit
2) If waveoff is given or clearance is cancelled		
a) Reason and waveoff issued within 2 seconds of receipt of cancellation, option not given	12**	35
b) Proper missed approach transmission used	11	15

* Resides in PV00
**Safety error

TABLE 38. PV11, OVER LANDING THRESHOLD COMPOSITE

Controller Action	Bit of PV11	Partial Credit
A. Over landing threshold		
1) Transmission given	7*,1	20
2) Given within ±1 second of the target contacting the landing threshold point	2	20
B. Final course position		
1) Given within 3 seconds of "over landing threshold"	3	20
2) Position correct (including "over" for "on" position)	4	20
3) "Over" is used correctly	5	20

*Resides in PVC0

TABLE 39. PV12, HANDOFF AND ROLLOUT COMPOSITE

Controller Action	Bit of PV12	Partial Credit
A. Rollout instructions on full-stop landing		
1) Rollout instructions given	8*,1	40
2) Instructions issued 20-40 seconds after "over"	2	20
3) Radio frequency is released within 10 seconds after rollout instructions	3	20
4) Pattern controller is notified	4	20
or		
B. Handoff to the pattern controller made if aircraft is on low approach or touch-and-go, or executing a missed approach including lost communications		
1) Handoff is given	9*,5	40
2) Handoff is made within 30 seconds of:	6	10
Condition: Waveoff — Reference Point: Issuance of waveoff		
Condition: Low approach — Reference Point: Decision height		
Condition: Touch-and-go — Reference Point: Landing threshold		
3) Call sign correct	7	5
4) Button correct	9	5
5) If missed approach, range must be given to nearest 1/2 mile, else not	10, 11	10
6) Monitor frequency and ICS until pattern transmits "CS radar"	12	10
7) Release radio frequency	10*,13	10
8) Pattern ICS selected during handoff	14	10

*Resides in PV00

TABLE 40. PV13, NO-GYRO COMPOSITE

Controller Action	Bit of PV13	Partial Credit
A. Warn pilot		20
"Heading XXX" given if 1/4 mile elapses after a turn and less than a 2° change in course is observed	1	
B. Prepare for no-gyro		
1) No-gyro approach announced	11*,7	30
2) No-gyro approach announced if course correction is not taken within 1/2 mile	2	10
3) The announcement issued prior to 3/4 mile from the point at which warning was issued	3	10
C. Make 1/2 standard rate turns		
1) Transmission given	12*,4	10
2) Issued after begin descent, and no-gyro announcement	5	10
3) Transmitted only once	6	10

*Resides in PV00

TABLE 41. PV14, NO-GYRO HEADING CORRECTIONS

Controller Action	Array Element of PV14 Error Counter		Weighting Factor Applied to Percentage Error
A. Turn was in correct direction	1	4	.4
B. "Stop turn" issued	2		.4
C. If target enters zone 3 from zone 2, a heading correction given within 20 seconds	3	5	.2

TABLE 42. PV15, EMERGENCY WAVEOFFS

Controller Action	Bit of PV15	Partial Credit
A. Radar contact lost		
1) If target moves off the display or the display fails, waveoff issued	1**	50
2) Issued within 5 seconds	2**	25
3) Proper R/T used for type of approach	3	25
or		
B. Target not touching at decision height		
1) Target not touching when decision height message given and waveoff issued. (If touching, ok.)	10**	50
2) "Too low" message used if that was highest priority position, else by some "too ..." message. Correctness of "too..." message scored in PV09, A2.	11**	25
3) Proper R/T used for type of approach	12	25

**Safety error

TABLE 43. PV16, LOW ALTITUDE ALERT

Controller Actions	Bit of PV16	Partial Credit
. Low altitude alert		
1) Transmitted when target exceeds 1 target width per mile below glidepath	1	50
2) Issued within 5 seconds	2	50

TABLE 44. PV17, TRANSMISSION BREAK

Controller Actions	Bit of PV17(0)	Array Element of PV17 Error	Array Element of PV17 Counter	Weighting Factor Applied to Percentage Error
A. Mike unkeyed after "over"	1	1	3	.8
B. At least one break given subsequent to "do not acknowledge" and prior to 1 mile	2	2	4	.2

TABLE 45. PV18, TRANSMISSION RATE

Controller Actions	Bit of PV18(0)	Array Element of PV18 Error	Array Element of PV18 Counter
A. Transmission rate after "do not acknowledge" advisory: Not more than 5 seconds between advisories	2	2	3

TABLE 46. PV19, RADAR ALIGNMENT COMPOSITE

Controller Action	Bit of PV19	Partial Credit
A. Alignment check preparation		
1) Azimuth: servo down until center-line reflector appears	7	10
2) Elevation and range: servo left until touchdown reflector appears	8	10
B. Select ALIGN if alignment of		
1) Azimuth	1,2	20
2) Elevation	3,4	20
or		
3) Range	5,6	20
is needed; else not		
C. Reposition antennae		
1) Servo up until 1-mile mark is bisected by glideslope	9	10
2) Servo right until the 1-mile mark is bisected by azimuth cursor	10	10

A, B, and C must be performed sequentially or no credit is given.

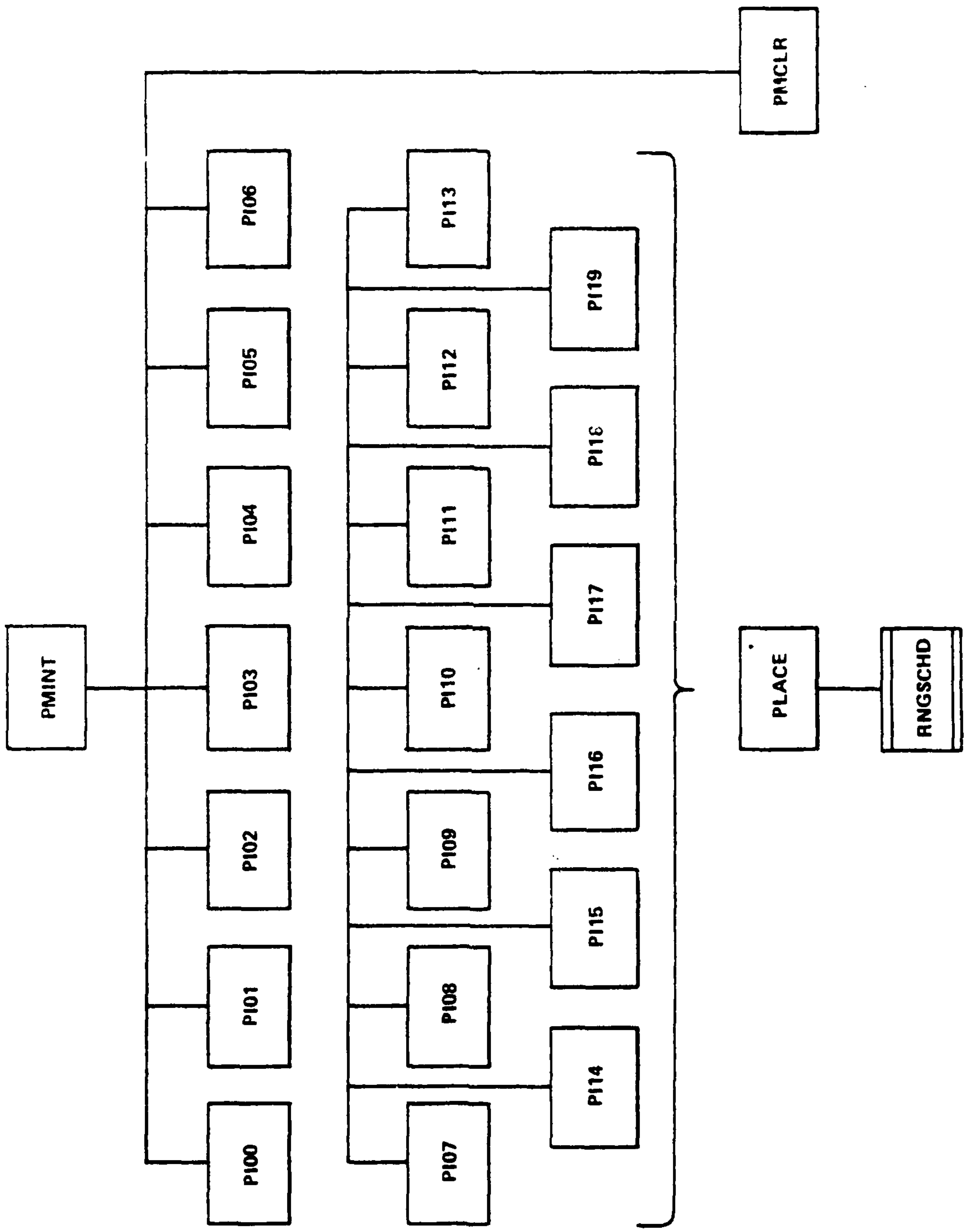


Figure 58. PMS Initialization Block Diagram

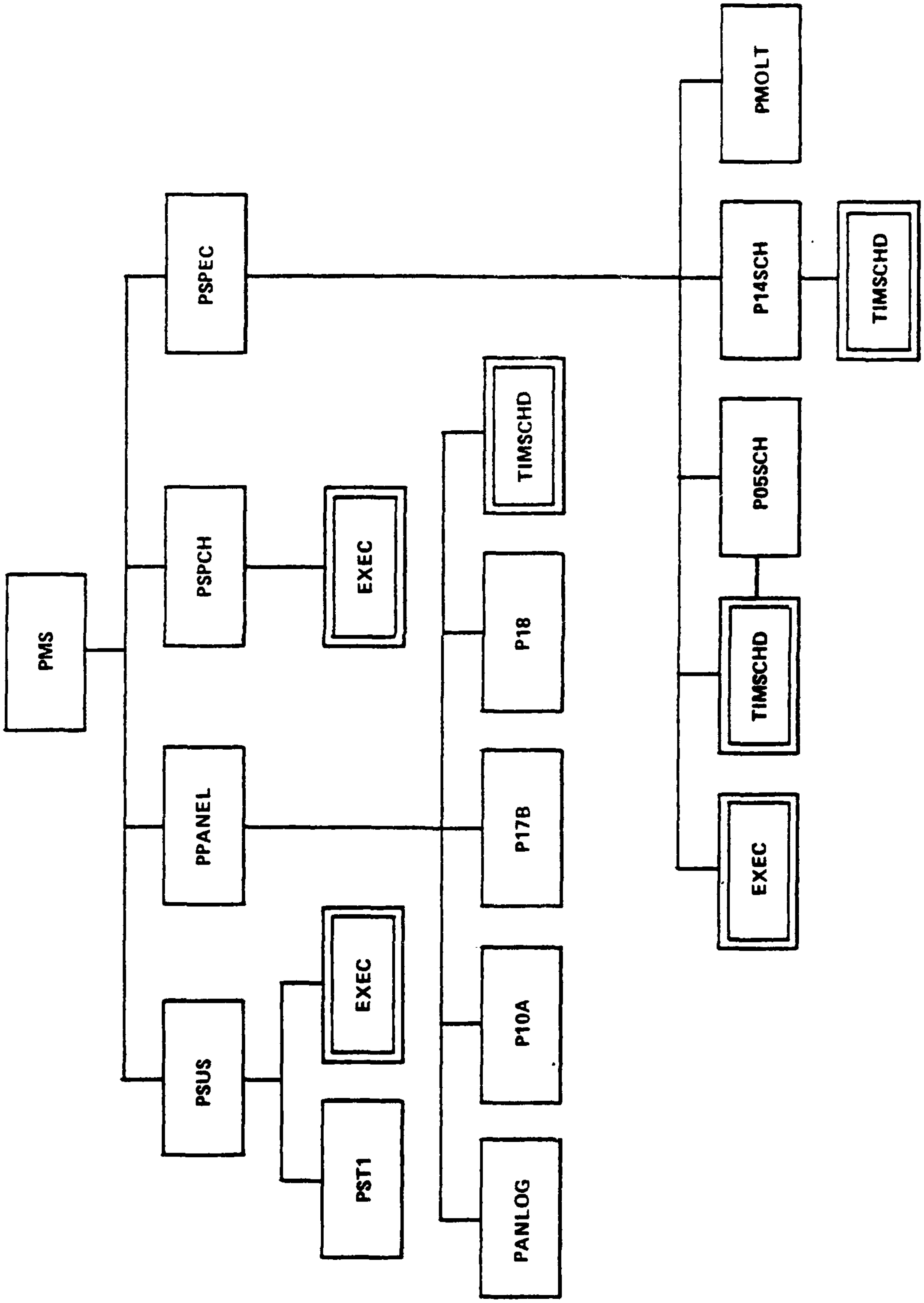


Figure 59. PMS Executive Routine Block Diagram

If the table slot referenced is not filled, no routine is executed for the event. Thus, unnecessary execution of event checking routines are avoided. In phase 2, PMS is brought in to score while the run is active. In phase 3, PMS scores after the run is complete. PMS does not need to know which phase is active to perform. PMS consists of a series of executives, performance variable error detection routines and omission checks. These routines act on a buffer stored in common. This common area, called SPACT, is a continually changing portrait of the state of the world during any part of a run. During a phase 2 run, this buffer is filled directly. In a phase 3 run, the records are written out to an activity file. After a phase 3 run, the routine PZEC is called to read from the activity file to the buffer in SPACT. It then repeatedly calls RDACT to distribute the contents of the buffer into the appropriate words in SPACT. Every time RDACT returns a 1 as its error argument, PZEC calls PMS. When PMS is called, it checks the type of activity record input and calls various routines to perform error checking. For example, PPANEL determines which panel button changed state and calls the error detection routines relating to the event. PPANEL uses a panel change mask to detect whether changes of interest have occurred. Bits in this mask are set by the PMINT routines. PPANEL will also schedule routines to check for the omission of events relating to a button being pressed. For example, if the waveoff light is flashing, the routine CKSWO is scheduled to check that the trainee gave the waveoff advisory properly. The routine PSPEC is called by PMS when a special record is written into SPACT. Special records include low altitude alert, and fast glidepath zone changes. PSPEC uses arrays PVSP1, PVSP2 and PVSP3 which are indexed by special event numbers to execute appropriate routines. PSPCH checks for proper advisories after the pilot gives a "weak but clear" advisory. When a speech advisory is read into SPACT, a more involved procedure is necessary because there are so many advisories to check. Advisories also need to be given in the correct order. The routine PSVS acts as executive to the speech advisories. There is an array called PVSVB indexed by the speech messages which contains the addresses of the error detection routines to be called for each message. A special assembly language program, EXEC, is called by PSUS with the indexed routine to be executed. Each routine called checks the validity of the current speech message and, if additional messages are required, schedules a routine to be executed later. This is accomplished in one of three ways. A routine can be time or range scheduled, or if a fixed order is desired, a bit is set in the word PVNEX to tell PSUS to call an additional routine.

Error recording. As mentioned previously, the performance routines detect errors. Error recording is accomplished by the routine PERRCHK. This routine is called whenever an error has been detected. Its job is to look up each error's index into the error explanation file. If it is phase 2, the error is explained to the student via the CRT, and the phase 2 executive is awakened to freeze the system. If it is phase 3, the error, time, index and PV number are recorded in the error file for later reporting during replay at the student's request. After all the records in the activity file have been read and acted upon, PZEC calls a routine called SCORE, which reads the bits set in the PV error words and calculates the final score. This score goes into the trainee's performance file in phase 3 runs and P-runs.

PERFORMANCE EVALUATION AND DATA FORMATTING ROUTINES. After PMS is through in phase 3 or after a P-run, the PMVs are scored and the results are made available to the trainee and instructor in a number of formats.

Performance Evaluation. Performance evaluation is accomplished by a set of routines marshalled by SCORE, which scores PV01 and PV02 and calls subroutines to evaluate the rest. Each PMV has a word (in array PVN) which indicates whether the word is being scored on this run. When a particular PMV is to be scored, this word is checked. If it is being scored, the score starts out at 100 and points are subtracted on the basis of Tables 27 through 46.

Data Formatting Routines. At various times it is possible to get written output on the progress of the trainee. There are several kinds of output which the instructor can request to get various degrees of overview of a student's performance. Type 1 output, Figure 60, gives the instructor a breakdown of the student's strengths and weaknesses on a task. Type 2 output, Figure 61, is a list of the scores attained on all the runs of a particular task. Columns of presently unscored areas are left blank. Type 3 output, Figure 62, is a detailed report of the student's performance on a single run with a description of the errors detected.

Because the instructor can ask for feedback on any completed run or task, it is necessary to provide a list of these runs. Figure 63 represents a sample list of the runs from which the instructor is asked to name a task or problem to be reported. Figure 64 shows the type 4 output, designed primarily for research use. A performance test printout is automatically produced. It is shown in Figure 65.

Finally, an off-line utility program, SFR, allows the instructor to print all or part of a trainee's files. Sample output is shown in Figure 66. This printout shows the student's last name, first name, identification number, and disk number along with the time and date that the printout was created. This is followed by a header which indicates the skill categories for the scores that follow each free practice task which is completed. An explanation of the skill category abbreviations is printed at the end of the printout. The rest of the printout indicates information about each individual training activity. The activities shown are:

a. Sign on, which indicates when the student signed on to the system.

b. Sign off, which indicates when the student signed off of the system.

c. Alignment, which shows the score given for the alignment procedure, if it was performed. No time is given for alignment since it begins immediately after sign on.

d. INIT V/T, which shows the times that the student entered the voice testing mode, and whether the student or the instructor initiated the test.

e. STOP V/T, which shows the time voice testing was terminated.

```
NAME: HARMON                    WILBUR          DATE: 3-13-1980   TIME:1958
                          PERFORMANCE ANALYSIS:
 STRENGTHS                   BORDERLINE                    WEAKNESSES

 HANDOFF                     GLIDEPATH POSITION/TREND      TURN TO FINAL
 RADIO CHECK                                               HEADING TRANSMISSIONS
 APPROACHING GLIDEPATH                                     RANGE CALLS
 AZIMUTH POSITION/TREND                                    EMERGENCY WAVEOFFS
 DECISION HEIGHT MESSAGE
 CLEARANCE REQUESTS
 LANDING THRESHOLD
 ROLLOUT OR  HANDOFF
 TRANSMISSION BREAK
 TRANSMISSION RATE

 STUDENT WAS ADVANCED TO PRESENT LEVEL AFTER COMPLETING  1 RUNS
 NO REMEDIATION NEEDED

 TOTAL SYSTEM TIME TO DATE:     16 HOURS AND    17 MINUTES
```

Figure 60. Type 1 Output: Strength and Weakness Report

NAME: HARMON WILBUR PERFORMANCE ON TASK: [illegible]

DATE: 3 6 1980 TIME: 1442

RUN#	HO*	RCh	TTF	AGP	HA	APT	GPT	RC	DH	CL	OLT	HR	NG	HC	EW	LAA	TB	TI
27	100	90	100		100	100		70		90								
28	100	80	100		100	50		[illegible]		95								
29	100	100	100		98	88		60		50								
30	100	100	100		92	80		90		100								
31	100	90	64		100	100		67		100								

*Explanations of abbreviations:
HO-Handoff; RCh-Radio check; TTF-Turn to final; AGP-Approaching glidepath; HA-Heading transmission; APT-Azimuth position and trend; GPT-Glidepath position and trend; RC-Range call; DH-Decision height; CL-Clearance; OLT-Over landing threshold; HR-Handoff or rollout; NG-No gyro; HC-Heading correction; EW-Emergency waveoff; LAA-Low altitude alert; TB-transmission break; [illegible]

Figure 61. Type 2 Output: The Scores for Each Problem in a Task

```
NAME: HARMON              WILBUR             PERFORMANCE ON PROBLEM 30, TASK: T03$32.03
DATE:  3-6 1980                              TIME: 1442

SUBJECT AREA                      SCORE  ERRORS
-------------------------------------------------------------------------------

ACCEPTING HANDOFF                  100   NONE

RADIO CHECK                        100   NONE

TURN-TO-FINAL                      100   NONE

HEADING TRANSMISSIONS               99   You gave a turn of 1 degree

AZIMUTH POSITION AND TREND          88   Incorrect position given in azimuth position call

RANGE CALLS                         80   Range call omitted
                                         Incorrect range used in range call

CLEARANCE                          100   NONE
```

Figure 62. Type 3 Output: Performance on a Single Problem

THE STATUS OF HARMON WILBUR AS OF 5-13-1980

TASK	TYPE	RESULTS	TASK NUMBER	PROBLEM RANGE
T02$22.03	NORMAL	NOT PASSED	14	1 - 8
T02$32.03	NORMAL	PASSED	18	9 - 14
T02$42.03	NORMAL	PASSED	22	15 19
T02$52.03	NORMAL	CONTINUED	26	20 24
T03$12.03	NORMAL	CONTINUED	31	25 - 25
T03$22.03	NORMAL	CONTINUED	35	26 - 26
T03$32.03	NORMAL	PASSED	39	27 - 31
T03$42.03	NORMAL	OUT OF PROBLEMS	43	32 - 37
T03$43.03	NORMAL	PASSED	44	38 - 42
T04$22.03	NORMAL	CONTINUED	51	43 - 47
T04$32.03	NORMAL	PASSED	55	48 57
T04$42.03	NORMAL	PASSED	59	58 - 60
T04$43.03	NORMAL	PASSED	60	61 - 63
T05$22.03	NORMAL	PASSED	66	64 - 83
T06$00.03	NORMAL	PASSED	68	84 93

Figure 63. Sample List of Completed Phase 3 Problems

```
            (1)                     (2)                      (3)                   (4)
NAME   JONES                       JOHN      PERFORMANCE ON TASK  101+03 01    MIN  5  MAX  16
              (5)                                        (6)
DATE   2- 8-1980                             TIME  1400

(7)RUN#  HO*  RCh  TTF  AGp  HA  APT  OPT  RCa  DH  C1  OLT  HR  NG  HC  EW  LAA  TB  TR
------------------------------------------------------------------------------------------
   (8)      (9)        (10)  (11)             (12)
   7     1-18-1980     0040   NR         ADAPTED -
      (13)  (14)  (15)  (16)  (17)                   (18)  (19)  (20)  (21)  (22)  (23)
      138    2     SI    FS    C2                    180    10     5     3     W

    (24)87    96    84
```

*Explanations of abbreviations:
HO=Handoff; RCh=Radio check; TTF=Turn-to-final; AGp=Approaching glidepath; HA=Heading transmissions; APT=Azimuth position and trend; OPT=Glidepath positon and tren; RCa=Range calls; DH=Decision height; C1=Clearance; OLT=Over-landing-threshold; HR=Handoff or rollout; NG=No-gyro; HC=Heading corrections; EW=Emergency waveoff; LAA=Low altitude alert; TB=Transmission break; TR=Transmission rate

(1) Student's last name.

(2) Student's first name.

(3) Task file name.

(4) Minimum and maximum number of problems needed to complete this task.

(5) The date this printout was created.

(6) The time this printout was created.

(7) The skill category associated with scores in that column (see (24)).

(8) The run number. This run number corresponds to the number quoted in "PROBLEM RANGE" of the Phase 3 and 4 "Task Listing," (Figure 25).

(9) The date that this run was scored.

(10) The time that this run was scored.

(11) Pattern controller information:
"R" if the pattern controller released the frequency;
"NR" if the pattern controller did not release the frequency.

(12) Environmental factors adapted due to poor performance on previous runs. If adaptation occurred, the words "Aircraft," "Pilot," and/or "Wind" will be printed depending on the variables adapted.

(13) Aircraft type.

(14) Pilot type:
Numbers range from 1 to 6. 1 is the best pilot, 6 is the worst.

(15) Starting aircraft position:
"SA" for short approach,
"RB" for right base,
"SI" for straight in,
"LB" for left base.

(16) Approach type:
"FS" for full stop,
"LA" for low approach,
"TG" for touch-and-go,
"E" for emergency,
"NG" for no gyro.

(17) Clearance type:
"N/A" for not applicable,
"C3" for clearance given at first request,
"C2" for continue, then clear at 2 miles,
"N" for clearance not given,
"WO" for waveoff,
"CX" for clearance given, then cancelled.

(18) Mean wind direction.

(19) Mean wind speed in knots.

(20) Mean gust speed in knots.

(21) Mean antigust speed in knots.

(22) Wheels down information:
"P" for given by pilot, not given by controller,
"PC" for given by pilot, given by controller,
"W" for not given by pilot, given by controller,
"NW" for not given by pilot, not given by controller.

(23) Gyro failure during final approach:
"GF" if gyro fails, else blank.

(24) The scores given for each skill category scored

Figure 64. Type 4 Output: Expanded Task Summary Report

PERFORMANCE RUN SUMMARY REPORT

NAME: HICKLIN

DATE: 8-16-1979 TIME: 1855

	REC	SPKR	RANGE	TIME	ADVISORY
	8	PTN	9.5	0	Position 4 Hand off Right base
	9	PTN	9.5	0	Marine 687 A6 Low approach Button 1
	19	GCA	9.5	11	Position 4 roger.
***				11	REFERENCE # 1 Handoff not acknowledged within 10 secs of issuance
	21	PTN	9.5	15	Marine 687 After completing Low approach Climb and maintain 1500
	22	PTN	9.5	15	Turn right heading 2 7 0
	23	PTN	9.5	15	Over
	26	PLT	9.5	27	Roger
	30	PTN	9.5	28	Marine 687 Turn right heading 150
	31	PTN	9.5	28	Over
	34	PLT	9.5	35	Roger
	45	GCA	8.6	68	Marine 687 Radar button 1.
***				78	REFERENCE # 2 Proper frequency not selected for radio contact

Figure 65. Performance Test Report

AD-A087 190 LOGICON INC SAN DIEGO CA TACTICAL AND TRAINING SYSTE--ETC F/G 17/9
GROUND CONTROLLED APPROACH CONTROLLER TRAINING SYSTEM (GCA-CTS)--ETC(U)
JUN 80 G BARBER, J BOLLENBACHER, D BREWTON N61339-77-C-0162

UNCLASSIFIED NAVTRAEQUIPC-77-C-0162-3 NL

3 OF 8
AD A087190

```
HARMON     WILBUR     258196081               D207         STUDENT FILE READER
DATE:  3-11-1980                              TIME: 2015   ACTIVITY FROM  3-6-1980 TO  3-6-1980
                NO*    RCh    TTF    AGe    HA    APT   GPT   RCa   DH    CT    OLT   HR   NG   HC   EW   LAA   TB   TR

3-6-1980   12:13:05   SIGN ON
           12:16:56   T02$51.01        PASSED
           12:23:47   T02$52.01        PASSED
           12:23:51   T02$52.02        CHALLENGED
           12:25:38   T02$52.03        OVERRIDDEN-F    5 RUNS       NO TIMEOUTS
            97    98     96      -      98      -       --    --    --    --      --      --    --    --
           12:51:35   T02$52.03
           12:53:02   T03$00.01        PASSED
           12:53:38   T03$11.01        PASSED
           13:23:39   T03$12.01        PASSED
           13:23:48   SIGN OFF
           13:41:10   NEW R/T          (INSTRUCTOR)        54    55    64
           13:42:58   T03$12.02        CHALLENGED
           13:43:26   T03$12.03        OVERRIDDEN-F    1 RUN        NO TIMEOUTS
           100    90     84      --     98   100   --   --   --    --    --    --    --    --    --    --    --
           13:55:05   T03$12.03
           13:55:54   T03$21.01        PASSED
           14:08:16   T03$22.01        PASSED
           14:08:19   T03$22.02        CHALLENGED
           14:08:40   T03$22.03        OVERRIDDEN-F    1 RUN        1 TIMEOUT
           100   100     90      --     96    75   --    77   --    --    --    --    --    --    --
           14:27:14   SIGN ON
           14:27:29   T03$22.03
           14:28:18   T03$31.01        PASSED
           14:41:49   T03$32.01        PASSED
           14:41:52   T03$32.02        CHALLENGED
           14:42:15   T03$32.03        PASSED          5 RUNS       NO TIMEOUTS
           100    92     92      --     99    85   --    72   --    87    --    --    --    --    --    --
           14:53:54   SIGN OFF
           15:11:46   SIGN ON
           15:37:47   T03$32.03
           15:38:49   T03$41.01        PASSED
           16:07:06   T03$42.01        PASSED
           16:07:11   SIGN OFF
           16:23:16   SIGN ON
           16:23:19   T03$42.02        CHALLENGED
           16:24:22   T03$42.03        NOT PASSED (TOO FEW PROBLEMS IN TASK FILE) 6 RUNS     NO TIMEOUTS
            99    89     79      --     96   100   --    70   --    62    --    88    --    --
           16:54:09   INIT V/T         (INSTRUCTOR)
           16:55:53   STOP V/T         (INSTRUCTOR)
           17:09:05   T03$42.03
           17:09:13   T03$43.03        PASSED          5 RUNS       NO TIMEOUTS
            92    88     86      -     100    94   --    88   --    77    --    --
           17:46:59   T03$43.03
           17:47:01   SIGN OFF
```

*Explanations of abbreviations:
HO=Handoff; RCh=Radio check; TTF=Turn to final; AGe=Approaching glidepath; HA=Heading
transmissions; APT=Azimuth position and trend; GPT=Glidepath positon and trend; RCa=Range calls;
DH=Decision height; CT=Clearance; OLT=Over-landing-threshold; HR=Handoff or rollout;
NG=No-gyro; HC=Heading corrections; EW=Emergency waveoff; LAA=Low altitude alert; TB=Transmission break; TR=Transmission rate

Figure 66. Off-line Printout of Student Files

f. NEW R/T, which shows the time NEW R/T activity began, the phrases retrained, and whether the student or the instructor initiated the training.

g. The time and date each training task was begun, the final disposition of the task, the number of approaches attempted during that task, the number of timeouts that occurred during that task, and if the task was free practice, the average scores attained for each skill category, or "--" if that skill category was not scored.

INTERPROCESSOR COMMUNICATION

GCA-CTS resides on two computers linked by the interprocessor bus. CPU 1 contains primarily the simulation and speech generation software. CPU 2 contains display and speech recognition software. Communication between these two systems is accomplished by the interprocessor bus, or IPB. The IPB is accessed as if it were a peripheral. Each computer in GCA-CTS has an IPB output routine, an IPB input routine, and one or more routing routines. A block diagram is displayed in Figure 67. The output routines are written in assembly language to make it possible to pass a variable length number of arguments to them. These routines are reentrant.

Besides a list of data, the output routines can accept two other types of input, text strings or an array. One of the arguments passed to IPBOUT* (where * = 1 or 2) indicates the type of format used. Upon entering IPBOUT*, a check is made to ensure that the number of arguments passed is correct. Each call to IPBOUT* contains an IPB identification code, and an array of legal arguments is kept in common, indexed by codes. If the format is a string, 80 characters are allowed, terminated by a carriage return, making the total number of characters 81. A list of the interprocessor identification codes exists in Appendix G. If the number of arguments for a particular code is incorrect, its identification code and the number of arguments passed are written to the bugs file by an error routine and control returns to the calling routine. IPBOUT* cannot distinguish between zero, one and two arguments. After this check is made, the arguments passed are organized in a standard form that can be understood by the IPB input routines. That form requires the argument to be ordered such that the argument with the lowest address is the first to be received on the other side. The number of arguments for the input routines to expect is also passed. The routines which receive arguments are all written in Fortran. IPBIN* (where * = 1 or 2) stores each group of messages in a buffer. Then these routines task routing routines to perform the functions indicated by the identification code. In some cases, where the function to be performed is not time consuming, IPBIN* will handle processing. There is one routing routine, TASKOUT, on CPU 1 and three on CPU 2, TALKOUT, LOOKOUT, and LOKFORWARD. This is because there is more interprocessor traffic going fr ~ CPU 1 to CPU 2. The codes are divided so that if several groups of arguments are sent, a task will not be delayed except where it is dependent on a similar task occurring first. For example, if several display messages are passed with a speech message, the display messages will be processed by a routing routine as they come in. The speech message, however, is not dependent on display and will therefore be processed in a different routing routine.

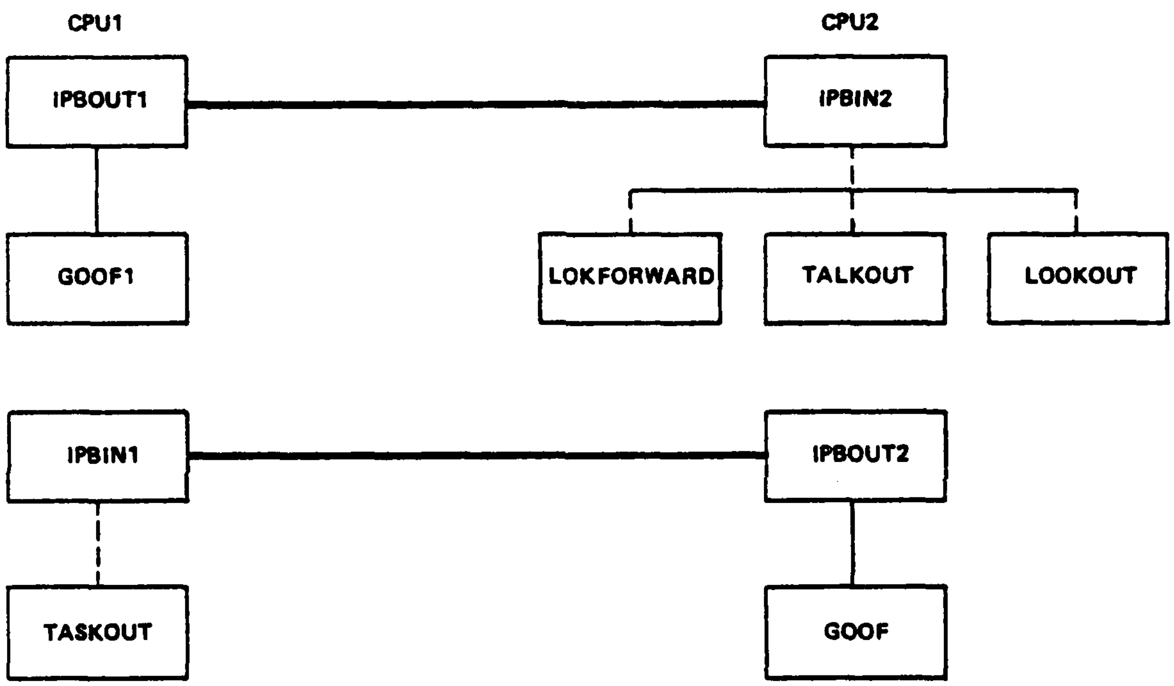


Figure 67. IPB Processing

CPU 2 has the added task of maintaining communication between foreground and the IPB. A routine called LOKFORWARD waits for the foreground buffer area to be free and then passes a new message to be processed by foreground.

KEYBOARD CONTROL

The keyboard on the GCA-CTS system is yet another way for the trainee to communicate with and influence the training process. There are two distinct keyboard software areas: keyboard input processing and keyboard support routines. Figure 68 is a block diagram of these routines.

KEYBOARD INPUT ROUTINES. There is a keyboard listening task in each computer whose job it is to read input from the keyboard, and display the name of the key pressed on the CRT. If the key is not one of those recognized by the training system, a question mark is printed on the screen. If the key pressed was MENU or the console interrupt toggle (shift MENU), processing takes place immediately. In all other cases, the valid key is processed by routines on the instructor side. If the key was pressed on the trainee side, the key is sent across the IPB. There are three routines on the instructor side to handle the processing of previously input keys. KPROC calls MENU to update the legal keys and routes the keys to KSTUD, if the key was pressed on the student keyboard or KTEACH, if the key was pressed on the instructor side or instructor functions were active on the student keyboard. The routine MENU is called whenever a GCA-CTS key is pressed. Once validity has been checked, the key is used as a switch in a special routine called DISPATCH to route it to the proper piece of code for processing. In effect, DISPATCH acts as a Fortran-assigned GO TO statement. It has the advantage, however, that it takes less space and can act upon a virtually unlimited number of items. Many of the keys are processed on CPU 1 with some special function keys requiring processing on CPU 2. If a key needs processing on CPU 2, a special message code is assigned to it and it is sent across the IPB to SKPRO on CPU 2. There the processing is completed.

KEYBOARD SUPPORT ROUTINES. The following subsections describe the details of these routines.

Introduction. A number of options are available to the instructor and student to modify the GCA-CTS training process. These options are designed to provide an uncomplicated, fairly natural interaction between instructor, student and computer using the keyboard and CRT for the interface. Those which require special routines are described in the following paragraphs.

Instructor Keyboard. The layout of the instructor keyboard is shown in Figure 69. The key names were chosen to be as descriptive of the key's function as possible within the maximum limits of 3 lines containing 4, 5, and 4 characters, respectively. The double width OVERRIDE key is an exception.

The functions of the keys shown are summarized in Table 47 and described in more detail as follows.

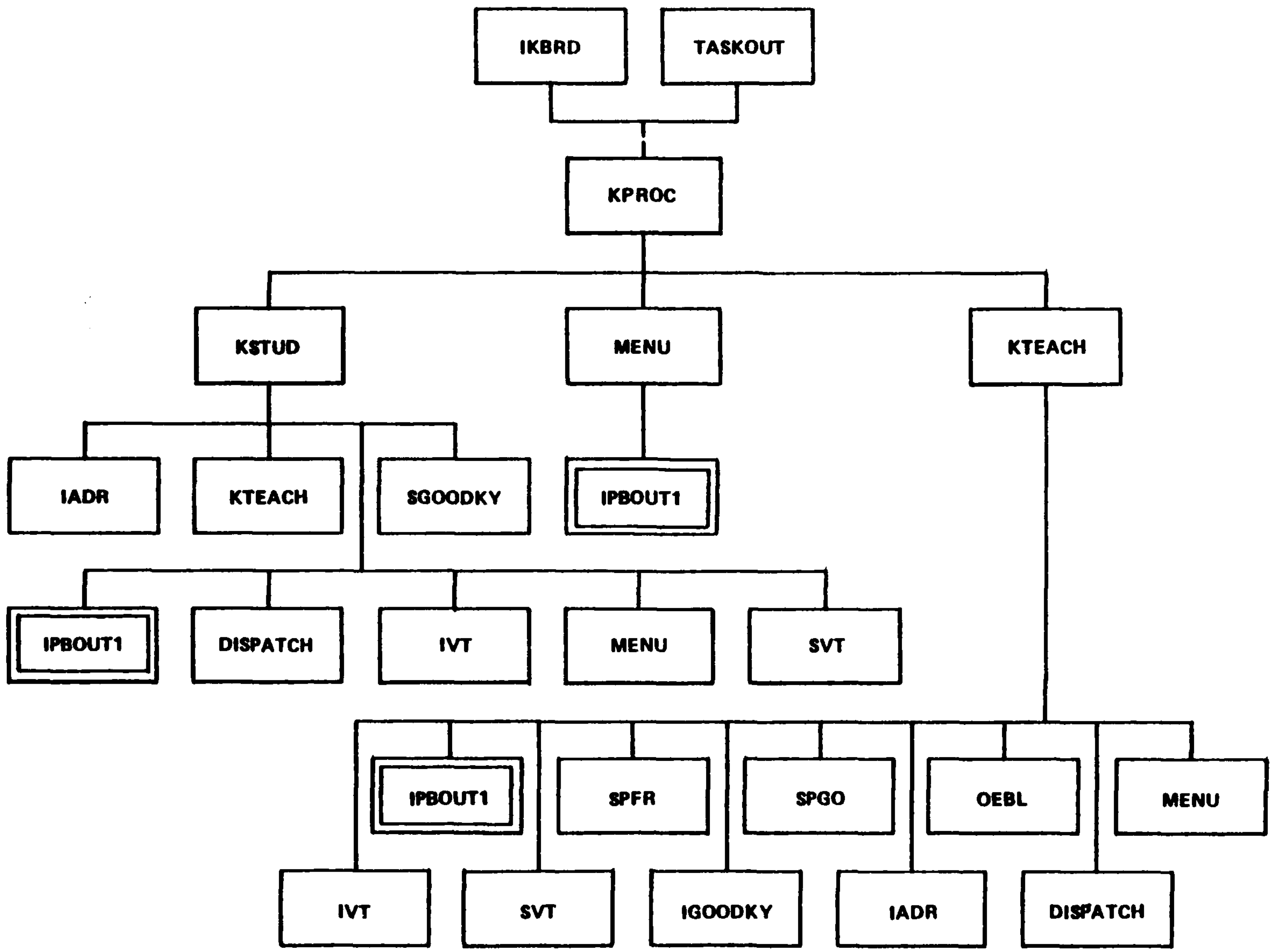


Figure 68. CPU 1 Keyboard Input Routines Block Diagram

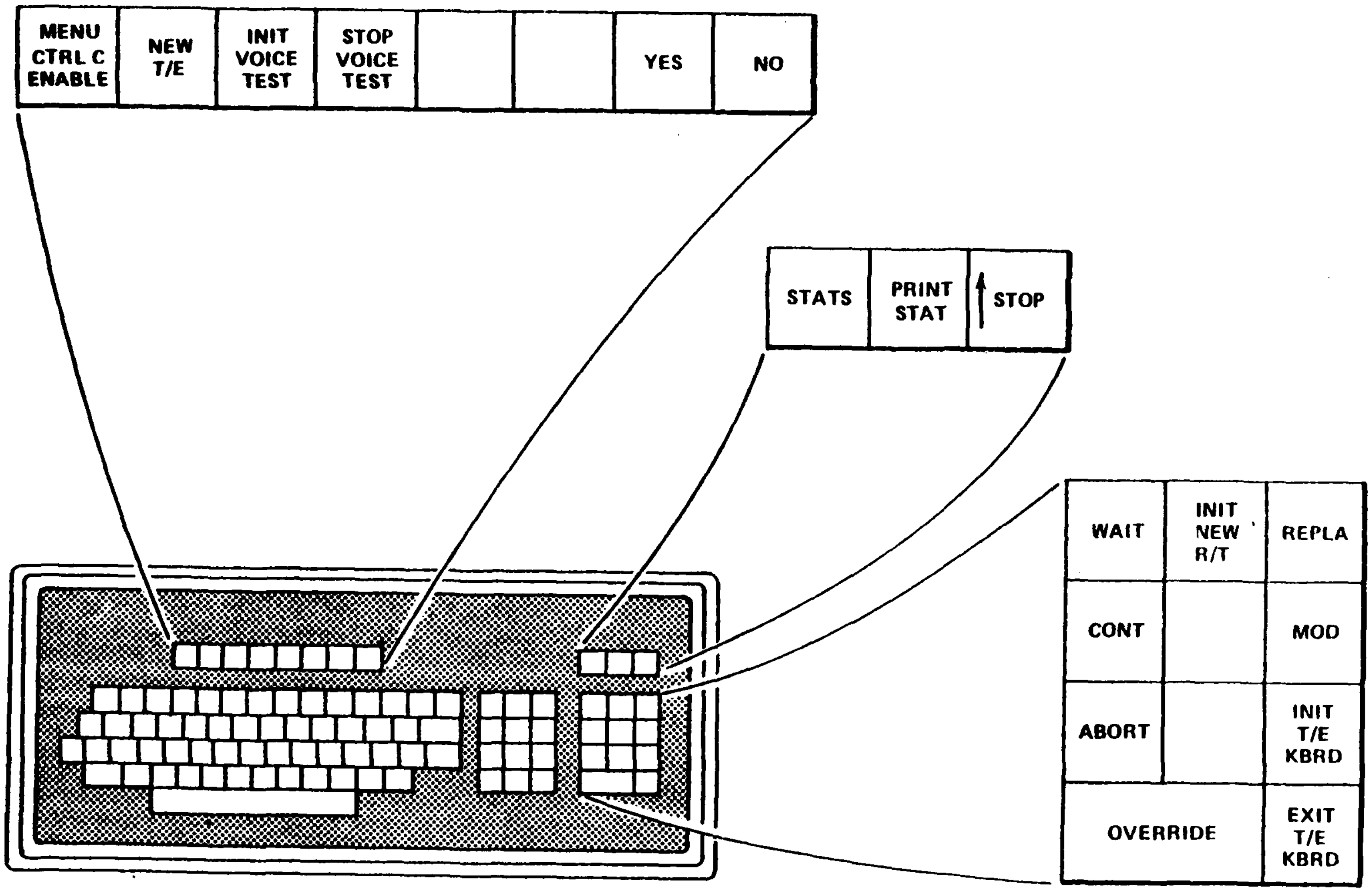


Figure 69. Instructor Keyboard Layout.

TABLE 47. FUNCTIONS OF KEYS AT INSTRUCTION STATION

Name	Octal Code	Function	Active
MENU	036,161	Displays on the CRT the legal keys for the current situation.	Always
NEW T/E*	036,162	Initializes new trainee files.	During Demo
INIT VOICE TEST	036,163	Causes the system to enter the speech validation mode at the conclusion of the present exercise. In the validation mode, the system will attempt to echo the spoken phrase.	After sign on
STOP VOICE TEST	036,164	Terminates speech validation.	After INIT V/T
YES	036,167	Used for responses to queries.	After queries
NO	036,170	Used for responses to queries.	After queries
STATS	036,171	Displays student status information on the CRT.	After sign on
PRINT STAT*	036,172	Provides detailed hard copy status reports.	Always
↑STOP (shift)	036,153	Causes the GCA-CTS program to terminate. Both processors return to the CLI.	Always
WAIT	067	Temporarily stops or freezes a demo or phase 3 run.	Demo, phase 3
CONT	064	Continues a run suspended by a WAIT, continues training after ABORT.	After WAIT or ABORT
ABORT	061	Stops the current run.	Phase 2, 3
OVERRIDE*	060	Allows the instructor to override GCA-CTS' problem selection.	After sign on
INIT NEW R/T*	070	Causes the speech data collection mode to be started after the completion of the present run.	After sign on
↑MENU	036,141	A debug option. By default, CTRL C is disabled. Pressing this key enables it. A subsequent press again disables CTRL C.	Always

TABLE 47. FUNCTIONS OF KEYS AT INSTRUCTION STATION (CONT)

Name	Octal Code	Function	Active
REPLA*	071	Causes replay of student's performance run after completion of the present run.	Always
MOD*	066	This key invokes the replay file editor which corrects any misrecognitions in the replay file. Training is suspended during this operation.	Always
INIT T/E KBRD		Activates the instructor functions on the trainee keyboard.	Always
EXIT T/E KBRD	056	Deactivates instructor functions on the trainee keyboard.	After INIT T/E KBRD

*Special purpose routine temporarily suspends normal keyboard processing.

Providing a Menu. Since not all keys are active or legal all the time, a key is provided called MENU which will display the menu or set of selections which is currently available. This key, like many others, is described as always active. This means that the system will respond to the key whenever the keyboard processing routine is in control. Certain keys cause the keyboard processing routine to be suspended while special purpose routines take over to handle user dialog. These keys are shown with an asterisk by their names. Thus, for example, the menu key, while described as always active in the table, would not be available for use after NEW T/E was pressed until the dialog involved in introducing a new trainee to the system was complete.

The code used to generate the menu can be found on both sides of the GCA-CTS system. Different menus are generated for each side, trainee and student. The menu is an array of words, one for each key on the student's and the instructor's keyboards. Each bit within the word corresponds to an episode in the GCA training system. Thus, there is a bit for demonstration, phase 1, P-run and so forth. The routine MENU determines the bit which describes the present state of the world. For each word in the array, if the bit location is set to 1, the key corresponding to that word is legal. When the MENU key is pressed, the names of all keys with the legal bit set will be printed on the CRT.

New Trainees. The instructor may introduce a new trainee to the system whenever a trainee is not signed on by using the NEW T/E key. This key causes the program NEWTE to be executed. NEWTE creates student files, including student performance and scratch files and fills the performance file with time and date of creation. The files are created on the removable disk in a directory with a special three-letter name created by GCA-CTS, based on the trainee's initials.

Statistics. There are two keys which cause summary of a student's progress. A brief summary of the current student's progress may be requested, to be typed on the CRT, by pressing STATS. The key PRINT STAT invokes the special routine PRNTIT, which provides a hardcopy printout in several different formats as desired by the instructor. There are several formatting routines used to accomplish the various output forms.

Initiating New Radio Terminology. The key INIT R/T causes the speech data collection mode to be started after the completion of the present run. This key should only be used by experienced users of the GCA training system, and appropriate precautions must be taken to ensure effective use. The instructor's guide has detailed information on its use. Its purpose is to collect input feature patterns (IFP) for a previously trained phrase. The user selects the phrase and the number of repeats (n). The student is prompted to repeat the phrase the requisite number of times. The new speech data replace the oldest n repeats, and a new voice reference pattern is made when speech data collection is complete. The user is then asked if more data are to be collected. If not, he is given the options to either INIT VOICE TEST, or CONTinue with the interrupted exercise. The routine INITRT starts the voice data collection, and prompts the user when necessary. There are routines on CPU 1 and CPU 2 which allow the instructor to initiate data collection on either side. The implementation avoids the necessity of heavy text transmission across the IPB.

Overriding Current Tasks. The key OVERRIDE initiates a dialog which allows the instructor to override the default problem selection to a limited extent. There are three options: 1) repeat a previous task immediately; 2) repeat a previous task when the present task is complete; and 3) terminate the present phase 2 or 3 task and initiate the next sequential syllabus task. In all cases, "task" here refers to an entry in the syllabus file which may contain many problem specifications.

The first option takes effect at the end of the present run or end of the present phase 1 exercise. Unless the present run results in scores that would normally have permitted advancement to the next task, the entire task which was interrupted will have to be repeated after the task the instructor selects is completed. This insures continuity in the student file.

The second option takes effect when the present task is complete. The instructor's selection is inserted prior to the selection automatically made by GCA-CTS. In neither of these is remedial training selected on the basis of the results of tasks selected by the instructor.

The third option allows the instructor to override the GCA-CTS assessment of the trainee's progress and to initiate the next sequential syllabus task despite the fact that the ordinary criteria for advancement have not been met. This likewise overrides any renedial exercise which might have been selected by the system.

This key is also available at the trainee station. If it is used there during phase 1, the initiation of the dialog is suspended until the phase 1 training is complete. A brief explanaton of the delay appears on the screen.

At present, the ABORT key must be pressed prior to the OVERRIDE key if a run is in progress.

Obtaining a Break File. By default, CTRL A and CTRL C will be disabled to prevent inadvertent program terminations. Pressing the shift key and the MENU key will toggle the CTRL C enable flag on and off. The routine to cause this is an assembly language routine named after the system call it makes, OEBL.

System Responses. Special YES and NO keys are provided for responses to system queries. A routine called GRESP determines the answer to each question and returns to different portions of the calling routine dependent on the answer given.

Replaying a Run. After performance runs the instructor has the opportunity to view the run with the use of the REPLA key. The instructor has a choice of two different types of replay. The first is a replay of both display and student responses with no acknowledgement of student error. This is primarily a research feature which will enable the instructor to make an independent evaluation of the student's performance. The second will freeze on errors and give the rule which applies to each error.

If the student is signed on, the REPLA request will take effect after the completion of the present run. The routine KREPLA determines the replay the instructor requests and causes it to take place.

Modifying the Replay Buffer. If the instructor perceives speech recognition errors during the performance run, he can modify the error file to correct them with the key MOD. At the end of the P-run, a copy of the student's error file is output to the printer. Using this printout and the replay file editor, he may correct misrecognized phrases and insert phrases where a sound was heard but not understood. When this task is completed the error file will be rescored with the new information. Training is suspended during this editing operation. If MOD is used while an exercise is in progress, the editor routine does not get control until the exercise is completed.

The routine to accomplish this is called MODIFY. First, it finds out the name of the trainee whose file is to be changed and makes his directory the default directory. If the directory could not be found, MODIFY quits. MODIFY then calls GAMOD which prints out a summary of the run if the instructor wants it. Then, as long as changes are still wanted, it asks which record is to be altered and calls SUBMODIFY with it. SUBMODIFY breaks down the message into components (e.g., whether "correction" was applied) and determines whether each component needs changing. If so, the instructor is asked for the correct form.

Student Keyboard. The student controller can also use the keyboard and CRT to enhance GCA training. The eleven special function keys located across the upper portion of the keyboard will be used for student keys. The keys available to him are dependent in part on the phase of training. Some of the instructor functions are also available on the student keyboard as shown in Figure 70, and the keys are described briefly in Table 48.

Trainee Signon. When a student wishes to begin training he presses HELLO. This initiates a dialogue to determine the student's name. The system ensures that he has been given a student file and that the file is available.

The routine HELLO on CPU 2 searches the removable disk for the student's file and, if present, readies it for training. If no file is found, the user is told via the CRT. CPU 1 is notified when the student has successfully completed signon.

Trainee Signoff. When the student wishes to take a break he presses BYE. This will alert the training system to stop after the current lesson or run is completed and to start the demonstration phase. The routine SIGNOFF on CPU 1 initiates shutdown.

Proceeding through a Task. When a line of text appears on the CRT, or when a run is over, the trainee is given time to read the text or to examine the display. This is done with the NEXT key. A routine called GETNEXT is called from the executives when a NEXT is required. A timeout occurs if the trainee does not respond.

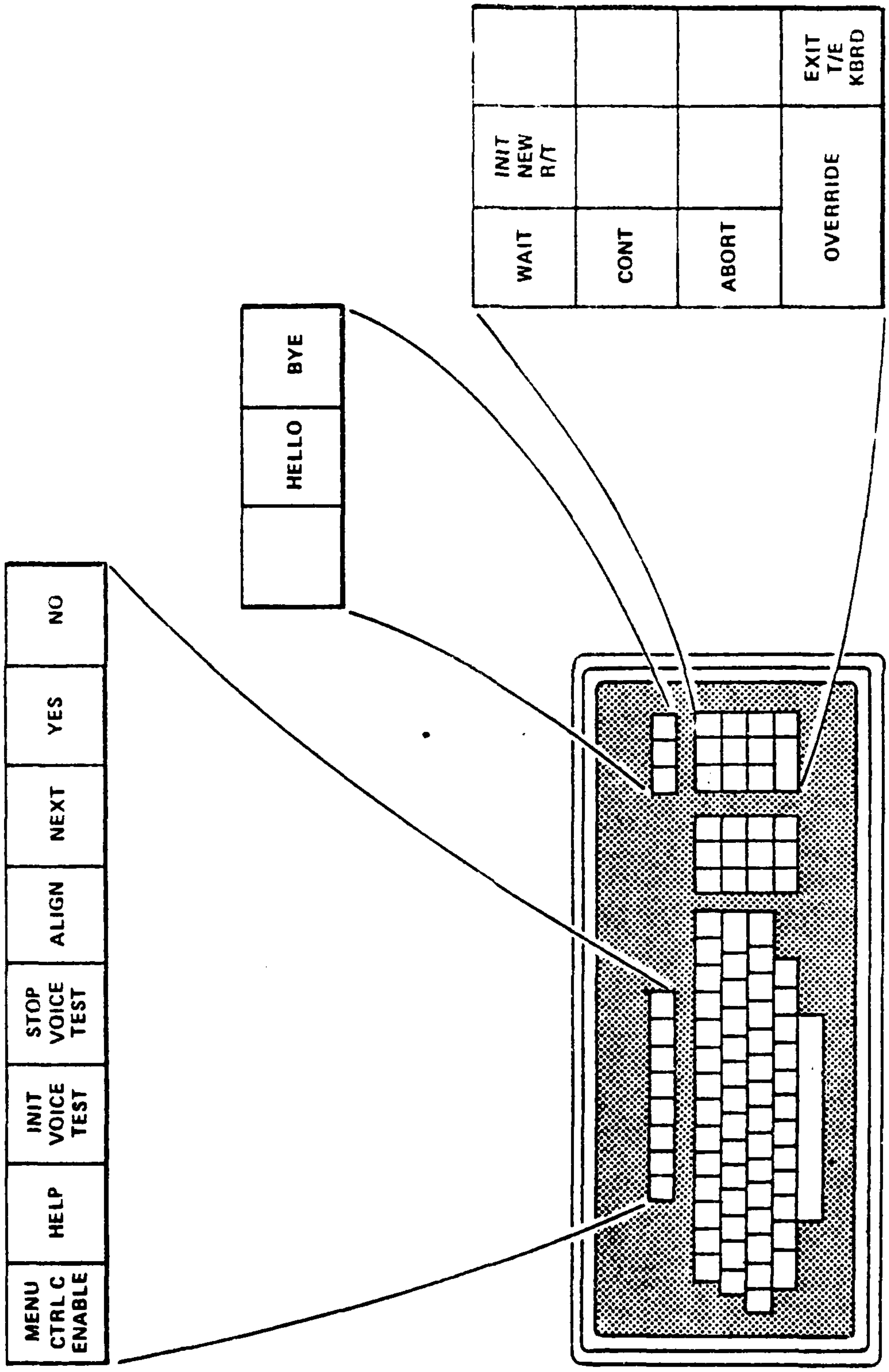


Figure 70. Trainee Keyboard Layout

TABLE 48. FUNCTIONS OF KEYS AT TRAINEE STATION

Name	Octal Code	Function	Active
MENU	036,161	Displays on the CRT the legal keys for the current situation.	Always
HELP	036,162	Displays a request for assistance on the instructor console.	After sign on
INIT VOICE TEST	036,163	Causes the system to enter the speech validation mode at the conclusion of the present exercise. In the validation mode, the system will attempt to echo the spoken phrase.	After sign on
STOP VOICE TEST	036,164	Terminates speech validation.	After INIT VOICE TEST
ALIGN	036,165	Sets centerline range and touchdown reflectors into proper aligrment.	After sign on
NEXT	036,166	Continues with the next frame of the lesson.	After queries
YES	036,167	Used for responses to queries.	After queries
NO	036,170	Used for responses to queries.	After queries
HELLO*	036,171	Initiates student sign-on procedure.	Demo
BYE	036,172	Terminates the session at the completion of the current problem. Demo will be started.	After HELLO
WAIT	067	Temporarily stops a demo or phase 3 run.	After INIT T/E KBRD
CONT	064	Continues a run suspended by a WAIT.	After WAIT or ABORT and INIT T/E KBRD
ABORT	061	Stops the current run.	Phase 2,3 after INIT T/E KBRD
OVERRIDE*	060	Allows the instructor to override GCA-CTS' problem selection.	After INIT T/E KBRD and HELLO

TABLE 48. FUNCTIONS OF KEYS AT TRAINEE STATION (CONT)

Name	Octal Code	Function	Active
INIT NEW R/T*	068	Causes the speech data collection mode to be started after the completion of the present run.	After HELLO
†MENU	036,141	Toggles CTRL C enable on and off.	Always
EXIT T/E	056	Deactivates instructor functions on the trainee keyboard.	After INIT T/E KBRD

*Special purpose routine temporarily suspends keyboard processing.

Aligning the Display. Ordinarily, at the beginning of the workshift, the controller checks the alignment of the PAR display. If the display is misaligned, the controller notifies the technicians to fix it. Because this facility is not available to GCA-CTS, an ALIGN key is provided to allow the trainee to correct misalignment.

TRAINEE AND INSTRUCTOR PANEL PROCESSING

The trainee and instructor panels are defined as an interrupt device. This device is attached to the RDOS interrupt structure by PANON and is removed by PANOFF.

The interrupt service routine (PINDR) services all panel interrupts. It builds and outputs the corresponding values to set the lights, etc., and sets the corresponding logical values. When PINDR receives an interrupt from the panel, it gets the state of the panel and exclusive ORs with the previous panel state. This produces a word whose bits indicate changes in the panel. The bit pattern is then used to determine what subroutine to call to handle the change.

Outputs to the panel are handled by PANOUT. When PANOUT is called, it sets the DOA and DOB words and then starts device 24 (the panel) so that the interrupt service routine can output the new values. This implementation makes it unnecessary for PANOUT to issue an INTDS, and also keeps the panel-related output to the activity file centralized.

During scoring (when panel changes are coming from the activity file instead of the panel) PANLOG is called to set the logicals. During REPLAY, it is necessary that the state of the SUPER/ICS displays not be replayed because these lights always reflect the present state of the panel. REPAN is designed to handle this.

APPENDIX A

MODULE SPECIFICATIONS

INDEX

INITIALIZATION ROUTINES

Title:	BLOCK1
Source file:	BLOCK1.FR
Description:	This is the CPU 1 block data program which initializes common variables for the simulations.
Classification:	Blockdata
Period:	None
Language:	F
Activated/called by:	N/A
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	BLOCK2
Source file:	BLOCK2.FR
Description:	This is the CPU 2 block data program.
Classification:	Blockdata
Period:	None
Language:	F
Activated/called by:	N/A
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	BLOCKF
Source file:	BLOCKF.FR
Description:	This is the CPU 2 foreground block data program.
Classification:	Blockdata
Period:	None
Language:	F
Activated/called by:	N/A
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	OLNM
Source file:	OLNM.FR
Description:	This simple routine constructs the overlay file name based upon the save file name. There are identical copies in CPU 1 and CPU 2.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	START1, START2
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	Array containing save file name.
Output arguments:	On return, input array has overlay file name.
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine is very simple minded. It assumes the user has retrieved the save file name via COMARG, and thus that it ends with ".", ".SV", or a null. It further assumes that the input array is large enough to hold the overlay file name with its extension. It will overwrite core if this condition is not met.

Title:	PATCH
Source file:	PATCH.SR
Description:	GCA-CTS load on call table causes MAC to overflow, hence a work around was devised which requires that the number of load on call routines be computed at runtime. This routine accomplishes this.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	START1
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	RTINIT
Source file:	RTINIT.FR
Description:	This routine initializes the range and time scheduled routines.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PB23SUB, P1AC, PZEC
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	RUNIT
Source file:	RUNIT.FR
Description:	This provides Phase 3 initialization for CPU 1.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PB23SUB
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	IER : Fortran error code
Files created/changed:	RPLDSP, RPLACT, RPLSPH, IDVFILE RPPDSP, RPPACT, RPPSPH, PIDVFILE
Files referenced:	None
Notes:	None

Title:	START1
Source file:	START1.FR
Description:	This is the CPU 1 startup routine.
Classification:	.MAIN task
Period:	None
Language:	F
Activated called by:	Fortran initializer
Cancelled by:	Self
Activates/calls:	SYSINIT, TZEC, PATCH, OLNM
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVKEY
Input arguments:	None
Output arguments:	None
Local variables:	I1 : array containing save and overlay file names I2 : array for switches from COMARG
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	START2
Source file:	START2.FR
Description:	This is the CPU 2 background startup routine.
Classification:	.MAIN task
Period:	None
Language:	F
Activated called by:	Fortran initializer
Cancelled by:	Self
Activates/calls:	TUNIT, SKBRD, OLNM
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	IER : error code I1 : array for overlay file name I2 : array for global switches
Files created/changed:	None
Files referenced:	CTSB.OL
Notes:	None

Title:	STARTF
Source file:	STARTF.FR
Description:	This is the foreground executive.
Classification:	.MAIN task
Period:	None
Language:	F
Activated/called by:	Fortran initializer
Cancelled by:	Stop message from CPU 1
Activates/calls:	PICUP, IMAGES, SERVO, CLOKF, OKTOUSEMEGATEK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	BXTIM
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	IER : error message from Fortran calls MSG : dummy message location IDUM : dummy function argument
Files created/changed:	FBUGS2
Files referenced:	None
Notes:	None

Title:	SYSINIT
Source file:	SYSINIT.FR
Description:	This is the CPU 1 initialization routine. It starts listening tasks for the keyboard, IPB and panel.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	START1
Cancelled by:	N/A
Activates/calls:	IKBRD, PANON, IPBIN1, OEBL, SPNIT, SPDMP, SPBUF, IPBOUT1, PANEL
IPB ID's used:	IDIMAGES
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	I : loop index IER : error code INPUT : IPB sync message
Files created/changed:	None
Files referenced:	Opens system files
Notes:	Caution! This routine must be loaded after SDBF.

Title:	TUNIT
Source file:	TUNIT.FR
Description:	This is the CPU 2 background initializer. It starts the tasks which are always active and establishes communication with CPU 1.
Classification:	Subroutine
Period:	None
Language:	F
Activated called by:	START2
Cancelled by:	N/A
Activates/calls:	SPINIT, IPBIN2, OEBL, CKCMN
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	OUTPUT : IPB test pattern IER : error code ITMP : IPB input temporary
Files created/changed:	None
Files referenced:	None
Notes:	None

TRAINING EXECUTIVE

Title:	ACSET
Source file:	ACSET.FR
Description:	This routine sets up the A/C call sign, speed and radio frequency.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PB23SUB, F1ACINIT
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	ACVERT
Source file:	ACVERT.FR
Description:	This routine changes the card inputs for all phases into the proper units and sets up the additional flags and variables which need to be initialized before each run.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PZDEMO, PZ23, F1ACINIT
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	LCLAA : logical, T if pilot causes a low altitude alert LCFLG : indicates if inputs are specified by zones or in feet and miles =1: inputs in feet and miles =2: inputs by zone Below used only on restricted flights LCLAZ : left azimuth zone LCRAZ : right azimuth zone LCLEZ : lower elevation zone LCUEZ : upper azimuth zone
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	ADAPT
Source file:	ADAPT.FR
Description:	This routine is called by P3TRM to suggest changes to reduce the difficulty of a problem based on the student's past performance. These changes are in the form of slower aircraft, better pilots, and milder wind conditions. When one of the suggested changes is made, it is noted in the student files.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	P3TRM
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	SUM
Notes:	– Scores range from 100 (perfect) to 0 (all wrong) with -1 indicating that skill was not scored. – This routine assumes that TZLSUM (the number of completed syllabus tasks) is initialized to zero when a student signs on.

Title:	CRSTUFE
Source file:	CRSTUFE.FR
Description:	This creates a text file of student feedback and directs CPU 2 to type it.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	P3TRM
Cancelled by:	N/A
Activates/calls:	IPBOUT1, STHELP
IPB ID's used:	IDSTUDSTATS
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	OLD : logical used to find out what to print I : loop index IER : error argument RUNS : temp
Files created/changed:	NCSTFE
Files referenced:	NCSCR
Notes:	None

Title:	DEMO
Source file:	DEMO.FR
Description:	DEMO conducts all types of approaches during system idle time. It provides for a smooth transition to alignment checking when the student signs on. It also serves to demonstrate system capabilities.
Classification:	Subroutine
Period:	None
Language:	F
Activated called by:	TZEC
Cancelled by:	N/A
Activates/calls:	IBPOUT1, P3BSUP, PZDEMO, SRMON, PI19
IPB ID's used:	IDSERVO
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	DEMOPROBS
Notes:	None

Title:	DESCRPROB
Source file:	DESCRPROB.FR
Description:	This routine explains the nature of a Phase 2 or Phase 3 problem to the student.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PB23SUB
Cancelled by:	N/A
Activates/calls:	IPBOUT1, GETNEXT
IPB ID's used:	IDCRT
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ITMP : GETNEXT input argument I : loop index
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	DIE
Source file:	DIE.FR
Description:	This routine kills GCA-CTS.
Classification:	Subroutine
Period:	None
Language:	F
Activated called by:	DWAIT, P1AC, P1DIS, P1INIT, P1PRM, P1SEQ, P1TXT, P1VDC, P1WAI, SHFSTOP, SINON, TZEC
Cancelled by:	N/A
Activates/calls:	SPFR, IPBOUT1, PANOFF, SPOFF
IPB ID's used:	IDDIE
Routines scheduled:	None
Cancels:	Everyone, sides 1 and 2
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	ERRTEST
Source file:	ERRTEST.FR
Description:	This routine checks Fortran errors.
Classification:	Subroutine
Period:	None
Language:	F
Activated called by:	PHAZ23, PZ23, RDTILNOTCO, RTZEC
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	IER : Fortran error code
Output arguments:	CODE : GCA-CTS error code
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	FTHSET
Source file:	FTHSET.FR
Description:	This routine sets up a table of turn headings for use by the model controller.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PB23SUB, P1PRM
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	FTHSET should not be called until all initialization changes in aircraft type, wind heading, wind speed, and starting position have been made (currently after ACSET).

Title:	GETANS
Source file:	GETANS.FR
Description:	This routine waits for an input from the trainee keyboard. If the trainee does not respond, a timeout occurs within a given time period and the trainee prompt is repeated. If a second timeout occurs, the routine tells the calling program to signoff the trainee.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	YORN, GETNEXT
Cancelled by:	N/A
Activates/calls:	IPBOUT1, TIMEOUT, TSKERRDLY
IPB ID's used:	IDCRT
Routines scheduled:	None
Cancels:	TIMEOUT
Mailboxes used:	None
Events referenced:	EVKEY
Input arguments:	LCMSG : the question to be answered N : dummy argument, no longer used LCREPLY : the student's reply LCWAIT : the number of seconds before timeout
Output arguments:	NEXT : the next task to perform, or -1
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	GETNEXT
Source file:	GETNEXT.FR
Description:	This routine prompts the student to type "next", then waits for that "next". Any special requests will be processed, and the system will sign off the trainee after two consecutive timeouts.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PHAZ23, P3TRM, DESCRPROB, KREPLAY, P2RUN, SC19
Cancelled by:	N/A
Activates/calls:	GETANS
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	LCWAIT : the number of seconds before timeout
Output arguments:	NEXT : the next task to perform, or -1
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	LOGRT
Source file:	LOGRT.FR
Description:	LOGRT keeps a record of all NEW R/T activity including time and date of activity, and the phrase numbers retrained.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	INIT2RT
Cancelled by:	None
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	WHO : the value of KBINST PHRASE : the phrase number retrained or 0 to indicate that this is an initialization of a new NEW R/T
Output arguments:	None
Local variables:	IDATE : the date ITIME : the time
Files created/changed:	LOG.RT
Files referenced:	LOG.RT
Notes:	The phrase number 0 is used to delimit records.

Title:	LOGVT
Source file:	LOGVT.FR
Description:	LOGVT keeps a record of all time spent in the Voice Test Mode of GCA-CTS.
Period:	Subroutine
Language:	F
Activated/called by:	VOICTST
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	CODE : 0 if voice test is beginning, 1 if voice test is ending WHO : the value of KBINST
Output arguments:	None
Local variables:	TRIED : a logical flag that is set true if we have tried to create file LOG.VT
Files created/changed:	LOG.VT
Files referenced:	None
Notes:	None

Title:	MOVIT
Source file:	MOVIT.SR
Description:	This routine moves a block of data in core using a block move instruction.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	SHUFFLE
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	P23SUB
Source file:	P23SUB.FR
Description:	This routine performs initialization for phases 2 and 3.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	P2RUN, P3RUN, PZ3B, PZDEMO
Cancelled by:	N/A
Activates/calls:	IPBOUT1, RNGSCHD, SPIN, PB23SUB, CLOK, APENIT, RZEC, PANOUT, FEED
IPB ID's used:	IDTIME, IDDIE
Routines scheduled:	GYROKILL
Cancels:	None
Mailboxes used:	None
Events referenced:	EVPHZ
Input arguments:	None
Output arguments:	None
Local variables:	IER : error code
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	P2FRZ
Source file:	P2FRZ.FR
Description:	This routine causes the system to freeze on errors in phase 2.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PERRCHK
Cancelled by:	N/A
Activates/calls:	IPBOUT 1, SUSTRM
IPB ID's used:	IDTIME
Routines scheduled:	None
Cancels:	CLOK, FEED, ENDFEED
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	IER : error code
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	P2RNSTOP
Source file:	P2RNSTOP.FR
Description:	This routine stops all of the run related tasks for the Phase 2 executives.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	P2RUN
Cancelled by:	N/A
Activates/calls:	SUSTRM, PANIT, RUNKILL
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	P2RUN
Source file:	P2RUN.FR
Description:	This routine starts and scores one Phase 2 run. It also selects the next task for the trainee.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TZEC
Cancelled by:	N/A
Activates/calls:	P23SUB, GETNEXT, P2RNSTOP, IPBOUT1
IPB ID's used:	IDCRT, IDFF
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVPHZ
Input arguments:	None
Output arguments:	NEXT : the next task TZEC should execute
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	GZPTRY must be set to zero to tell PZ23 to read the number of error free runs for the next problem.

Title:	P3BSUP
Source file:	P3BSUP.FR
Description:	This routine reads and decodes the information on Phase 3 multipossibility card sets and converts the data to cumulative percentages for problem selection.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	DEMO, PZ23
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	INBUF : array containing ASCII card image of the first card in the student file.
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	Problem files T##$##
Notes:	None

Title:	P3PBLK
Source file:	P3PBLK.FR
Description:	This routine writes out one record to the Phase 3 file.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	P3TRM
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	P3
Files referenced:	None
Notes:	None

Title:	P3RUN
Source file:	P3RUN.FR
Description:	This routine initiates one Phase 3 run. It also changes the student file pointer dependent on whether the student should advance, or should repeat the problem.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TZEC
Cancelled by:	N/A
Activates/calls:	RUNSTOP, P3TRM, SCHWRITE, P23SUB
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVPHZ
Input arguments:	None
Output arguments:	NEXT : the next task to initiate
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	P3TRM
Source file:	P3TRM.FR
Description:	This routine is used by Phase 3 executives to invoke scoring, and replay as requested.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	P3RUN, PZ3B
Cancelled by:	N/A
Activates/calls:	PZEC, SCHWRITE, YORN, REPLAY, SELECT, CRSTUFE, RPFOR, SHUFFLE, FOR2, ADAPT, P3PBLK, PANIT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	NEXT : the next task to initiate, or -1
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PB23SUB
Source file:	PB23SUB.FR
Description:	This routine calls the run-related tasks and performs the adaptation for Phase 3, 2, and P-run problems.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	P23SUB
Cancelled by:	N/A
Activates/calls:	ISAY, MODELINIT, RUNIT, ACTOUT, ACSET, FTHSET, PANIT, RTINIT, DESCRPROB, PMINT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVPHZ
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine has been broken into two parts (P23SUB, PB23SUB) for the sake of overlay allocation.

Title:	PHAZ23
Source file:	PHAZ23.FR
Description:	This routine reads Phase 2, 3, and P-run header cards.
Classification:	Subroutine
Period:	None
Language:	F
Activated called by:	TZEC
Cancelled by:	N/A
Activates/calls:	PZERR, PZSCREEN, PZTXT, SR1ST, SFSET, GETNEXT, RDTILNOTCO, ERRTEST
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	NEXT : next task to be executed
Local variables:	PMS : temporary storage for read ICARD : input buffer for card read
Files created/changed:	SCRATCH, student files
Files referenced:	Problem files
Notes:	None

Title:	PZ23
Source file:	PZ23.FR
Description:	This routine reads and initiates single possibility problems.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TZEC
Cancelled by:	N/A
Activates/calls:	P3BSUP, ERRTEST, ACVERT, SR1FIN, PZERR, RDTILNOTCOMNT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	NEXT : the next task to initiate
Local variables:	None
Files created/changed:	SCRATCH
Files referenced:	P3
Notes:	None

Title:	PZ3B
Source file:	PZ3B.FR
Description:	This routine directs a phase 3 multipossibility approach.
Classification:	Subroutine
Period:	None
Language:	F
Activated called by:	TZEC
Cancelled by:	N/A
Activates/calls:	RUNSTOP, PZSEL, P3TRM, ACVERT, P23SUB
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVPHZ
Input arguments:	None
Output arguments:	NEXT : next task to be performed
Local variables:	JUNK: dummy arguments
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PZDEMO
Source file:	PZDEMO.FR
Description:	This routine is the executive for demonstrations. It calls the routines to execute new approaches until a trainee signs on.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TZEC
Cancelled by:	N/A
Activates/calls:	PZSEL, ACVERT, P23SUB, RUNKILL, SINON, APE1NIT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	SRMON
Mailboxes used:	None
Events referenced:	EVPHZ
Input arguments:	None
Output arguments:	Next : the next task to be initiated
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PZERR
Source file:	PZERR.FR
Description:	This routine writes to the bugs file messages concerning errors in the problem files or student files. It kills CPU 2 and returns to the CLI.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PZ23, PHAZ23, RTZEC
Cancelled by:	N/A
Activates/calls:	IPBOUT1
IPB ID's used:	IDDIE
Routines scheduled:	None
Cancels:	IPBIN1
Mailboxes used:	None
Events referenced:	None
Input arguments:	CODE : type of user error IER : Fortran error code
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PZREQ
Source file:	PZREQ.FR
Description:	PZREQ handles the processing of special keyboard requests from the instructor and student.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TZEC
Cancelled by:	N/A
Activates/calls:	VOICTST, SR1FIN, SCHWRITE, MENU, STSK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVKEY
Input arguments:	None
Output arguments:	Next : the next task to execute
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PZSCREEN
Source file:	PZSCREEN.FR
Description:	This routine turns on the appropriate Megatek display pictures for Phase 2,3,4, and DEMO runs.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	DEMO, PHAZ23
Cancelled by:	N/A
Activates/calls:	IPBOUT1
IPB ID's used:	IDIMAGES, IDSERVO
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	LCERD : control for elevation display LCARD : control for azimuth display
Output arguments:	None
Local variables:	1CONOFF : used in call to IPBOUT1 to indicate if pictures are on or off
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PZSEL
Source file:	PZSEL.FR
Description:	This routine selects approach conditions based on information in a Phase 3 multi-possibility card set or a demonstration run specification file.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PZ3B, PZDEMO
Cancelled by:	N/A
Activates/calls:	SELNV
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	WINDVARIATION : temporary for condition described by common variables ENWVP, ENWSCT, ENGOCC WINDDIRECTION : average wind direction described by common variable ENWHT GUSTINESS : amount of fluctuation in wind speed described by common variables ENMGS and ENMGD WINDSPEED : average wind speed described by common variable ENMWS LCPOS : position of aircraft in terms of range, altitude, and heading CONDITION : speed of aircraft described by common variable ACTYP LCGYR : 2 : no gyro failure to occur, else not RUNWAY : 10 : runway visible, else not WHEELS : 1 : wheels are down, 2 - wheels are up, described by common variable PTWEEL SIGN : 1 : right of course for wind direction, -1 : left of course
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PZTXT
Source file:	PZTXT.FR
Description:	This routine displays a page on the student console.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PHAZ23
Cancelled by:	N/A
Activates/calls:	IPBOUT1
IPB ID's used:	IDTEXT
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVTXT
Input arguments:	None
Output arguments:	IER : 1 if text was displayed -1 if an error occurred
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	RDTILNOTCO
Source file:	RDTILNOTCO.FR
Description:	This routine reads the comment cards in the phase files and displays them at the instructor station CRT.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PHAZ23, PZ23
Cancelled by:	N/A
Activates/calls:	ERRTEST
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	DISPLAY: true if comment is to be written out
Output arguments:	ICARD: array for file data IER : error code CODE : error code BYTES : number of bytes in ICARD
Local variables:	None
Files created/changed:	None
Files referenced:	T**$**.02, T**$**.03, T**$**.04
Notes:	None

Title:	REMSEL
Source file:	REMSEL.FR
Description:	This routine selects the next remedial task if there is one.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RTZEC
Cancelled by:	N/A
Activates/calls:	IPBOUT1
IPB ID's used:	IDCRT
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	LCNAME : name of the remedial training file
Files created/changed:	None
Files referenced:	Remedial training file, SCRATCH
Notes:	None

Title:	RR1FIN
Source file:	RR1FIN.FR
Description:	This routine acts as an interface between a just completed task and the next task. It writes one record for each task to the student file and summarizes Phase 3 runs for the trainee.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	SR1FIN
Cancelled by:	Self
Activates/calls:	TFB, SUMPUT, SCHWRITE
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVPHZ
Input arguments:	WHY : reason for completion of task
Output arguments:	None
Local variables:	PHASE : present training phase P3PTR : pointer to Phase 3 problem block I : index in write statement SUMPTR : pointer to Phase 3 summary block, or -1 IER : dummy arg for error flag LCBUFF : latest record written to SR1 IDATE : array to be filled with present date ITIME : array to be filled with present time
Files created/changed:	SR1 : student file
Files referenced:	None
Notes:	None

Title:	RTZEC
Source file:	RTZEC.FR
Description:	The training system executive initiates individual phase executives as required.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TZEC
Cancelled by:	N/A
Activates/calls:	SCHREAD, IPBOUT1, YORN, REMSEL, PZERR, ERRTEST SR1FIN, SCHWRITE, SC19
IPB ID's used:	IDSERVO, IDCRT, IDFF, IDSKPRO
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	NEXT : next task to execute
Local variables:	ICARD : buffer for reading syllabus BYTES : returned byte count in RDLIN LCMIN : time of day in minutes ITIME : time of day hr:min:sec IDATE : date IER : error variable
Files created/changed:	SCRATCH
Files referenced:	Student records, syllabus
Notes:	None

Title:	RUNKILL
Source file:	RUNKILL.FR
Description:	This routine stops APE and model controller and insures that the last record is written to the replay file.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RUNSTOP, P1AC, P1PRM, P2RNSTOP
Cancelled by:	N/A
Activates/calls:	IPBOUT1
IPB ID's used:	IDTIME, IDDIE
Routines scheduled:	None
Cancels:	CLOK, FEED, ENDFEED, SAYIT, indirectly kills APE through BXCYC
Mailboxes used:	BXCYC
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This is an exact copy of the Phase 1 routine KILLRUN.

Title:	RUNSTOP
Source file:	RUNSTOP.FR
Description:	This routine stops all of the run related tasks for the Phase 2 and 3 executives.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	P2RUN, P3RUN, PZ3B, PZDEMO
Cancelled by:	N/A
Activates/calls:	RUNKILL, SUSTRM, PANIT, IPBOUT1
IPB ID's used:	IDIMAGES
Routines scheduled:	None
Cancels:	None
Mailboxes used:	BXSPH
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SCHINIT
Source file:	SCHINIT.FR
Description:	This routine initializes the scratch file for new trainees.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	NEWTE
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	SCRATCH
Files referenced:	None
Notes:	None

Title:	SCHREAD
Source file:	SCHREAD.FR
Description:	This routine reads one record of the scratch file.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RTZEC, GETDIR
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	I : record number of file to read
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	SCRATCH
Notes:	None

Title:	SCHWRITE
Source file:	SCHWRITE.FR
Description:	This routine writes one record to the scratch file.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RR1FIN, SGNOFF, PZREQ, P3TRM, P3RUN, RTZEC, SR1ST
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	I : record to write
Output arguments:	None
Local variables:	None
Files created/changed:	SCRATCH
Files referenced:	None
Notes:	An attempt has been made to group data in records according to when and how often it needs to be updated. Record 2 changes fastest.

Title:	SELECT
Source file:	SELECT.FR
Description:	This routine selects the new problem for the trainee. This problem may be a new task, or a remedial task. It also acts upon instructor overrides which request advancement, and writes the decision to the student files.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	P3TRM
Cancelled by:	N/A
Activates/calls:	SR1FIN
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	ADVANCE : dummy argument
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SELNV
Source file:	SELNV.FR
Description:	This routine selects environmental conditions randomly within the limit specified for this exercise.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PZSEL
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	CPCNT : array of cumulative percentages NELM : upper bound of CPCNT SEED : seed value for random number generator
Output arguments:	CONDITION : output index of condition selected in CPCNT
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SFSET
Source file:	SFSET.FR
Description:	This routine sets the PMV flags to cause scoring of specific events during a run.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PHAZ23
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SHUFFLE
Source file:	SHUFFLE.FR
Description:	Orders the activity replay file by time.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	P3TRM
Cancelled by:	N/A
Activates/calls:	MOVIT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	RPPACT, RPLACT
Files referenced:	None
Notes:	None

Title:	SR1FIN
Source file:	SR1FIN.FR
Description:	This routine starts the RR1FIN in node 3.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RTZEC, PZREQ, PZ23, SR1ST, TZEC
Cancelled by:	N/A
Activates/calls:	RR1FIN, TSKERRDLY1
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVPHZ
Input arguments:	I : dummy argument for RR1FIN
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SR1ST
Source file:	SR1ST.FR
Description:	This routine initiates the student performance files for a new problem.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PHAZ23
Cancelled by:	N/A
Activates/calls:	SR1FIN, SCHWRITE
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	P3
Notes:	None

Title:	SRMON
Source file:	SRMON.FR
Description:	This routine issues a verbal warning if student activates the servo while demo has an aircraft on final.
Classification:	Task
Period:	.5 seconds
Language:	F
Activated/called by:	DEMO
Cancelled by:	PZDEMO
Activates/calls:	GLIB, IPBOUT1
IPB ID's used:	IDSERVO
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	LCSETX : saves current Azimuth servo position LCSETY : saves current Elevation servo position
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	STSK
Source file:	STSK.FR
Description:	This routine starts a requested task which requires keyboard input and waits for its completion by termination of instructor input or timeout.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PZREQ
Cancelled by:	N/A
Activates/calls:	ZTIM, INITRT, OVERRIDE, IKBRD, IPBOUT1, PRNTIT, MODIFY, NEWTE
IPB ID's used:	IDCRT, IDFF
Routines scheduled:	None
Cancels:	ZTIM, OVERRIDE, NEWTE, INITRT
Mailboxes used:	None
Events referenced:	EVZEC
Input arguments:	TASKNAME : task entry OVLYNAME : overlay TID : task id
Output arguments:	None
Local variables:	STATUS : task status IER : Fortran error code
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SUMPUT
Source file:	SUMPUT.FR
Description:	This routine writes a Form 1 type summary of a student's performance on a Phase 3 task.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RR1FIN
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	MIN : minute the run occurred HOUR : hour the run occurred
Files created/changed:	FNFORM1
Files referenced:	None
Notes:	SUMPUT must be called after TFB, which averages the scores and puts them in the PVE array.

Title:	TFB
Source file:	TFB.FR
Description:	This routine calculates the average score attained by a student for each P.V. and writes out the averages to the student records.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RR1FIN
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	JUNK : temporary IDATE : date of score averaging ITIME : time of score averaging LCPTOT : array of PMV score totals, indexed by PMV
Files created/changed:	SUM : student record
Files referenced:	None
Notes:	None

Title:	TIMEOUT
Source file:	TIMEOUT.FR
Description:	This is a timeout task for keyboard input.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	GETANS
Cancelled by:	GETANS
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVKEY
Input arguments:	Seconds : time to wait
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	TSKERRDLY
Source file:	TSKERRDLY.FR
Description:	This routine is a delay routine used by CPU 2 routines when a task cannot be started immediately. There are identical copies in foreground and background.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	INIT2RT, IPBIN2, LOOKOUT, SERVUP, TALKOUT
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	TID : ID of task which cannot be started
Output arguments:	None
Local variables:	None
Files created/changed:	BUGS2, FBUGS2
Files referenced:	None
Notes:	None

Title:	TSKERRDLY1
Source file:	TSKERRDLY1.FR
Description:	This routine delays on task errors.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	ENDFEED, EXPLAIN, FEED, GETANS, GLBF, HOSAY, IKBRD, IMOFF, IPBIN1, P1AC, RSB, SR1FIN, SUCPH, TASKOUT
Cancelled by:	None
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	TASKID : ID of task which could not be started
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	TZEC
Source file:	TZEC.FR
Description:	This routine is the executive for all phases of training. It starts the demo, reads in problem cards and calls routines to start runs and sign the trainee off.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	START1
Cancelled by:	N/A
Activates/calls:	PZREQ, DEMO, PZDEMO, RTZEC, PHAZ23, PZ23, P2RUN, P3RUN, PZ3B, PHZ1, SR1FIN, SGNOFF, KREPLA, DIE
IPB ID's used:	None
Routines referenced:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	YORN
Source file:	YORN.FR
Description:	This routine sends the student a yes or no question and waits for a reply. If no reply is given before two timeouts occur, the system signs off.
Classification:	Subroutine
Period:	None
Language:	F
Activated called by:	RTZEC, KREPLAY, P3TRM
Cancelled by:	GETANS, Self
Activates/calls:	IPBOUT1, GETANS
IPB ID's Used:	IDCRT
Routines scheduled:	None
Cancels:	TZEC
Mailboxes used:	None
Events referenced:	None
Input arguments:	$1 : label for yes answer return $2 : label for no answer return LCMSG : array containing question to be asked N : # of characters in LCMSG
Output arguments:	NEXT : the next task to initiate or -1
Local variables:	LCREPLY : student response
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	ZTIM
Source file:	ZTIM.FR
Description:	This is the timeout task for TZEC.
Classification:	Task
Period:	None
Language:	F
Activated called by:	STSK
Cancelled by:	STSK
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVZEC
Input arguments:	NUMBER : number of seconds to delay
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

PHASE 1 INTERPRETER

Title:	DIGIN
Source file:	DIGIN.FR
Description:	This routine handles the collection of digitized speech samples.
Classification:	Subroutine
Period:	None
Language:	F
Activated called by:	P1PRM
Cancelled by:	N/A
Activates/calls:	IPBOUT1, SPIN, SPOUT, DWAIT
IPB ID's Used:	IDCRT, IDPRESENT
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVSPT
Input arguments:	IARG : 7 word integer array IARG(1) : 0 IARG(2) : phrase number IARG(3-7) : 0
Output arguments:	None
Local variables:	ITRY : number of tries NFLAG : used to inform digitizer that repeat was requested MSG : used to distinguish timeouts
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	DWAIT
Source file:	DWAIT.FR
Description:	This routine is used to keep track of timeouts, responses, etc. during the collection of digital voice samples.
Classification:	Subroutine
Period:	None
Language:	F
Activated called by:	DIGIN
Cancelled by:	N/A
Activates/calls:	RESPOND, TIMER, IPBOUT1, DIE
IPB ID's Used:	IDCRT
Routines scheduled:	None
Cancels:	RESPOND, TIMER
Mailboxes used:	BXFZ1
Events referenced:	None
Input arguments:	ITYPE : response type 1 : speech input 2 : keyboard entry ITIME : time to wait MSG : input request 1 : start tasks 2 : task already active
Output arguments:	MSG : output wait code 1 : EVSPN occurred 2 : "YES" entered 3 : "NO" entered 4 : timeout
Local variables:	IER : error code LMSG : temporary for MSG
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	F1ACINIT
Source file:	F1ACINIT.FR
Description:	This routine initializes APE variables in phase 1.
Classification:	Subroutine
Period:	None
Language:	F
Activated called by:	P1AC
Cancelled by:	N/A
Activates/calls:	ACVERT, ACSET, IPBOUT1
IPB ID's Used:	IDDIE
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	T**$**.01
Notes:	None

Title:	P1AC
Source file:	P1AC.PR
Description:	This routine is used to initialize, start, freeze and terminate APE (and model control if app).
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PHZ1
Cancelled by:	N/A
Activates/calls:	F1ACINIT, RTINIT, APENIT, RZEC, IPBOUT1, TSKERRDLY, RUNKILL, VARIMOD, DIE, CLOK
IPB ID's used:	IDSERVO, IDDIE
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVPHZ
Input arguments:	None
Output arguments:	None
Local variables:	IER : Fortran error code IDENT : identifier for type of action to perform
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	P1AZLR
Source file:	P1AZLR.FR
Description:	This routine is used to check for various wait conditions as defined in the phase 1 instruction file.
Classification:	Task
Period:	N/A
Language:	F
Activated/called by:	P1WAI
Cancelled by:	P1WAI, Self
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:rguments:	None
Mailboxes used::	BXFZ1
Events referenced:	EVKEY, EVVIN, EVVRO, EVSPT
Input arguments::	IARG : information on type of wait expected, keyboard, voice, etc.
Output arguments:	None
Local variables:	IER : Fortran error code
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	P1DIS
Source file:	P1DIS.FR
Description:	Phase 1 routine which initiates requested displays on the MEGATEK.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PHZ1
Cancelled by:	N/A
Activates/calls:	IPBOUT1, DIE
IPB ID's used:	IDIMAGES, IDMEGSTR
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	IDENT : type of display desired DISP1 : wind speed DISP2 : wind direction
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	P1END
Source file:	P1END.FR
Description:	P1END is used to close the instruction file and the digitized voice files on CPU 1 and to cause PLATEXT.FR to close the text file on CPU 2.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PHZ1
Cancelled by:	N/A
Activates/calls:	IPBOUT1
IPB ID's used:	IDTEXT
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	NVMARG : number of arguments to be sent across IPB. ITASK : type of task to execute on side 2 of IPB. Tnn$nn.01 (phase 1 instruction file) CANFILE, CIDFILE (digitizer files)
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	P1INIT
Source file:	P1INIT.FR
Description:	P1INIT is an initialization routine called upon entry to PHZ1.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PHZ1
Cancelled by:	N/A
Activates/calls:	IPBOUT1, PANIT, DIE
IPB ID's used:	IDTEXT
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	IER ; Fortran error code ISTAT : array containing file status after call to RSTAT I : loop index
Files created/changed:	None
Files referenced:	Txnn$nn.01 : text file for current run Tnn$nn.01 : phase 1 instruction file
Notes:	None

Title:	P1PRM
Source file:	P1PRM.FR
Description:	This routine initiates generation of prompts for phase 1.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PHZ1
Cancelled by:	N/A
Activates/calls:	VSPRES, MODELINIT, DIGIN, RUNKILL, SPOUT, RZEC, IPBOUT1, FTHSET, PANIT, RNGSCHD, FEED, PANOUT, DIE
IPB ID's used:	IDPRESENT
Routines scheduled:	GYROKILL
Cancels:	None
Mailboxes used:	None
Events referenced:	EVSPT
Input arguments:	None
Output arguments:	None
Local variables:	IDENT : type of prompt requested IER : Fortran error code I : loop index
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	P1RAD
Source file:	P1RAD.FR
Description:	This routine initiates servo positioning, alignment, and activity for Phase 1.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PHZ1
Cancelled by:	N/A
Activates/calls:	IPBOUT1
IPB ID's used:	IDSERVO
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	J : loop index IDENT : indicates activity to perform IVAR : alignment variable SERPOS : servo position
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	P1SEQ
Source file:	P1SEQ.FR
Description:	Phase 1 file sequence commands are handled by this routine. Skips, subroutine, and conditional command types are provided.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PHZ1
Cancelled by:	N/A
Activates/calls:	DIE
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	I : loop index J : loop index
Files created/changed:	None
Files referenced:	None
Notes:	Setting FZSKP to 0 via the instruction file command of "skip # of records" causes PHZ1 to loop forever at the current record #.

Title:	P1TXT
Source file:	P1TXT.FR
Description:	Phase 1 routine which causes text to be displayed on either instructor or student CRTs.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PHZ1
Cancelled by:	N/A
Activates/calls:	IPBOUT1, DIE
IPB ID's used:	IDCRT
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	TXTREC : logical page of text IDENT : text line destination
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	P1VDC
Source file:	P1VDC.PR
Description:	This routine initiates Phase 1 voice data collection.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PHZ1
Cancelled by:	N/A
Activates/calls:	IPBOUT1, MENU, DIE
IPB ID's used:	IDSPEECH
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVPHZ, EVSTP
Input arguments:	None
Output arguments:	None
Local variables:	IDENT : type of activity to perform
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	P1WAI
Source file:	P1WAI.FR
Description:	This routine initiates wait conditions for Phase 1.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PHZ1
Cancelled by:	N/A
Activates/calls:	P1AZLR, TIMER, DIE
IPB ID's used:	None
Routines scheduled:	None
Cancels:	P1AZLR, TIMER
Mailboxes used:	BXFZ1
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	IDENT : type of activity to perform
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PHZ1
Source file:	PHZ1.FR
Description:	PHZ1 is the training executive for Phase 1 instruction during which the proper use of radio terminology is taught while formulating student voice reference patterns.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TZEC
Cancelled by:	N/A
Activates/calls:	P1INIT, P1END, P1VDC, P1DIS, P1PRM, P1RAD, P1AC, P1WAI, P1SEQ, P1TXT, DIE
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	INSTR : integer instruction type IER : Fortran error code
Files created/changed:	None
Files referenced:	None
Notes:	The Phase 1 instruction file pointer 'FZPTR' always points to the current record. Normal progression through the file is caused by incrementing it by the 'SKIP' variable 'FZSKP'.

Title:	PLATEXT
Source file:	PLATEXT.FR
Description:	This routine displays instructional text on the student's CRT.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	LOOKOUT
Cancelled by:	Self
Activates/calls:	IPBOUT2
IPB ID's used:	IDAWAKE
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVTXT
	IDENT :
	1 - open file
	2 - display page # TEXT(1)
	3 - error
	4 - error
	5 - close file
Input arguments:	TEXT : text file name or page number
Output arguments:	IDENT : set to -1 when processing is complete
Local variables:	I : loop index
	LCOUNT : end of logical page pointer
	IER : error code
	IPOINT : logical page pointer
	INDEX : array for text to be output
	NAME : temporary array for text file name
Files created/changed:	None
Files referenced:	Instructional text files : TX**$**.**
Notes:	PLATEXT assumes the text files have been formatted by CONTEXT.

Title:	RESPOND
Source file:	RESPOND.FR
Description:	This routine waits for trainee responses in Phase 1.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	DWAIT
Cancelled by:	DWAIT
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	BXFZ1
Events referenced:	EVSPN, EVKEY
Input arguments:	ITYPE : response expected
Output arguments:	None
Local variables:	MSG : temporary for ITYPE
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	TIMER
Source file:	TIMER.FR
Description:	This is a timeout task used by Phase 1.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	DWAIT, P1WAI
Cancelled by:	DWAIT
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	BXFZ1
Events referenced:	None
Input arguments:	IWAIT : time to wait
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

REPLAY

Title:	ACDMP
Source file:	ACDMP.FR
Description:	This routine writes an activity block to RPLACT.
Classification:	Task
Period:	None
Language:	F
Activated called by:	ACTOUT
Cancelled by:	Self
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	RPLACT
Files referenced:	None
Notes:	Caution! This routine must have a lower priority than any of ACTOUT's callers.

Title:	ACTIVITY
Source file:	ACTIVITY.FR
Description:	This goes through a buffer from RPLACT and responds according to what was found in the first word of each record: 1 : call RPINITAC to initialize replay mode 2,3, or 7 : skip the record 4 : call REPAN to handle panel changes; set KYSPH 5 : call IPBOUT1 with servo information 6 : call GLIB with speech generator output.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	ATRPLY
Cancelled by:	Self
Activates/calls:	RPINITAC, REPAN, GLIB, IPBOUT1, SURPLY
Cancels:	None
IPB ID's used:	IDSERVO
Routines scheduled:	None
Mailboxes used:	BXACT
Events referenced:	None
Input arguments:	BUFFER : with 256 words from RPLACT
Output arguments:	MSG : 1 indicates a 2-word activity is on the buffer boundary 0 indicates there is more to be done -1 indicates that the last activity has be processed -n indicates that an error occured at time n
Local variables:	PTR : pointer into BUFFER BOX : temporary box for REC
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	ACTOUT	
Source file:	ACTOUT.SR	
Description:	This routine fills a buffer with activity file information. In Phase 3 it starts a buffer dump routine at the appropriate time.	
Classification:	Subroutine	
Period:	None	
Language:	A	
Activated/called by:	Caller	Activity Record Type
	PB23SUB	1
	SPDMP	2
	ACTSUS	3
	PANEL	4
	DONE	6
	APGP	7,1,12
	CKCRP	7,1,3
	CKGPP	7,1,8; 7,1,9
	ENDAPGP	7,1,11
	ENDFEED	7,1,2
	FEED	7,1,5
	IMOFF	7,1,6
	LOST	7,1,10
	LOW	7,1,4
	TGT50	7,1,1
	MILER	7,2
	DECK, MARKIT, OLT	7,3
	TOWER	7,4
	SUGYRO	7,5
Cancelled by:	N/A	
Activates/calls:	ACDMP	
IPB ID's used:	None	
Routines scheduled:	None	
Cancels:	None	
Mailboxes used:	None	
Events referenced:	None	
Input arguments:	Accepts inputs of two forms: FORM1 : 0, n, array of size n; FORM2 : ARG1, ARG2, ..., ARGN where n is a multiple of 8.	
Output arguments:	None	
Local variables:	None	
Files created/changed:	None	
Files referenced:	None	
Notes:	ACTOUT is reentrant and does not scramble the inputs even though they are moved to a common area. The only restriction is that ACDMP has a lower priority than other tasks in the system so that it cannot get control while a low priority task is still filling the middle of the buffer. If the first word of the record to be written is -1, ACTOUT causes the last activity file record to be output.	

Title:	ATRPLY
Source file:	ATRPLY.FR
Description:	ATRPLY tasks RDBUFF to read a block from RPLACT and calls ACTIVITY to go through that block a record at a time during REPLAY.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	REPLAY
Cancelled by:	REPLAY
Activates/calls:	RDBUFF, ACTIVITY
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	Passed and received as arguments, BXACT
Events referenced:	None
Input arguments:	CHNL : channel # for RPLACT
Output arguments:	CALLBOX : mailbox to indicate when done
Local variables:	MBOX1, MBOX2 : mailboxes to pass to RDBUFF BLNUM : pointer into RPLACT BLNUM1 : temporary buffer pointer for task reading into buffer 1 BLNUM2 : temporary pointer for task reading into buffer 2 BOX : temporary for REC MSG : message indicating activity is finished
Files created/changed:	None
Files referenced:	RPLACT
Notes:	None

Title:	ERINDEX
Source file:	ERIN.FR
Description:	This function retrieves an index from the error index file which points to the error explanation text in ERXFI.
Classification:	Function
Period:	None
Language:	F
Activated/called by:	FR301, FR304, FR912, FRREST, PERRCHK
Cancelled by:	N/A
Activates/calls:	None
Cancels:	None
IPB ID's used:	None
Routines scheduled:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	BIT : PMV bit WORD : PMV number
Output arguments:	ERIN : ERXFI record number
Local variables:	IER : error code RECORD : array which holds indicies for a PMV
Files created/changed:	None
Files referenced:	ERBLK
Notes:	None

Title:	ERLOOKUP
Source file:	ERLOOKUP.FR
Description:	Prints state-of-the-world info for EXPLAIN such as the correct call sign.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	REXPLAIN
Cancelled by:	N/A
Activates/calls:	None
Cancels:	None
IPB ID's used:	None
Routines scheduled:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	NUM : an index to tell it what info to give CHAN : channel number for output
Output arguments:	None
Local variables:	XMPS : real temp for range
Files created/changed:	None
Files referenced:	VOTEXT
Notes:	None

Title:	ERRHAN
Source file:	ERRHAN.FR
Description:	This handles errors found in ERRFI during the replay mode: removes user clock, kills digitized speech, calls EXPLAIN to output error, waits for student to respond that he is ready to continue, and restarts clock and digitized voice.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	REPLAY
Cancelled by:	N/A
Activates/calls:	EXPLAIN, SPFR, SPGO, RPCLOK
Cancels:	None
IPB ID's used:	None
Routines scheduled:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	INDEX : error index into ERXFI
Output arguments:	None
Local variables:	IER : error argument
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	EXPLAIN
Source file:	EXPLAIN.FR
Description:	This starts the task which explains errors during replay.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PERRCHK, ERRHAN
Cancelled by:	N/A
Activates/calls:	REXPLAIN, TSKERRDLY
Cancels:	None
IPB ID's used:	None
Routines scheduled:	None
Mailboxes used:	None
Events referenced:	EVEXPL
Input arguments:	INDEX : pointer into ERXFI
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	RADOUT
Source file:	RADOUT.FR
Description:	This accepts a buffer of 256 words from RPLDSP or RPPDSP and goes through it 8 words at a time, sending information to the display at the sweep rate during replay.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RDRPLY
Cancelled by:	N/A
Activates/calls:	IPBOUT1
Cancels:	None
IPB ID's used:	IDPKSRV
Routines scheduled:	None
Mailboxes used:	BXRPL
Events referenced:	None
Input arguments:	BUFFER : a block from RPLDSP
Output arguments:	MSG : informs caller that replay of display data is complete
Local variables:	PTR : a pointer into BUFFER BOX : a temporary to pass to REC
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	RDBUFF
Source file:	RDBUFF.FF
Description:	This reads a block from the channel passed to it into the buffer passed to it during replay and XMTs to the mailbox when its done.
Classification:	Subroutine and/or Task
Period:	None
Language:	F
Activated/called by:	REPLAY, ATRPLY, RDRPLY
Cancelled by:	Self
Activates/calls:	None
Cancels:	None
IPB ID's used:	None
Routines scheduled:	None
Mailboxes used:	Accepted as argument
Events referenced:	None
Input arguments:	CHNL : channel number to access file BLNUM : pointer to next block to be read
Output arguments:	BUFAD : filled w/block MBOX : 1 means found, -1 means not
Local variables:	RETNU : # of blocks read by RDBLK MSG : output message for output to caller = 1 : normal read = -1 : end of file encountered
Files created/changed:	None
Files referenced:	Input via CHNL
Notes:	None

Title:	RDERR
Source file:	RDERR.FR
Description:	This routine initializes ERR.CO common.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PERRCHK, PRNTIT, PZEC, SC19
Cancelled by:	N/A
Activates/calls:	None
Cancels:	None
IPB ID's used:	None
Routines scheduled:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	RDRPLY
Source file:	RDRPLY.FR
Description:	This causes the display information to be replayed by keeping the replay buffers full and calling the display data processing routine.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	REPLAY
Cancelled by:	REPLAY
Activates/calls:	RDBUFF, RADOUT
Cancels:	None
IPB ID's used:	None
Routines scheduled:	None
Mailboxes used:	Passed and received as arguments, BXRPL
Events referenced:	None
Input arguments:	CHNL : channel number for RPLDSP
Output arguments:	CALLBOX : mailbox to indicate when done
Local variables:	MOBX1, MBOX2 : mailboxes to pass to RDBUFF BLNUM : pointer into RPLDSP BLNUM1 : temporary pointer for task1 of RDBUFF BLNUM2 : temporary pointer for task2 of RDBUFF BOX : temporary for REC MSG : used to inform RDRPLY that RPLDSP is finished
Files created/changed:	None
Files referenced:	RPLDSP
Notes:	None

Title:	REPLAY
Source file:	REPLAY.FR
Description:	REPLAY tasks ATRPLY and RDRPLY and activates the digitized voice to replay student files. When it is time to report an error it waits until the speech digitizer is silent, then calls ERRHAN to freeze the system and explain the error.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	P3TRM, KREPLAY
Cancelled by:	N/A
Activates/calls:	IPBOUT, RDBUFF, ATRPLY, RDRPLY, SPOUT, ERRHAN, STRTPLY, RPCLOK
Cancels:	RDRPLY, ATRPLY, RPCLOK
IPB ID's used:	None
Routines scheduled:	None
Mailboxes used:	Accepted and passed as arguments, BXSPH
Events referenced:	EVERR, EVPHZ
Input arguments:	TYPE (of run): = 1 : no errors reported = 2 : errors reported
Output arguments:	None
Local variables:	RMBOX : mailbox to pass to RDRPLY to find out when its done AMBOX : mailbox to pass to ATRPLY to find out when its done MBOX : temporary to pass to REC IER : error argument ERBLNUM : pointer into ER or PER for next block to read ERRBUFF : to hold block read from ER or PER ERRPTR : pointer into ERRBUFF LASTERR : used to calculate length of time to wait to report next error
Files created/changed:	None
Files referenced:	ER, PER
Notes:	None

Title:	REXPLAIN
Source file:	REXPLAIN.FR
Description:	This explains errors during replay by outputting text messages and information about the state of the world.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	EXPLAIN
Cancelled by:	N/A
Activates/calls:	ERLOOKUP, PRSUS, IPBOUT1
Cancels:	None
IPB ID's used:	IDSTUDSTATS, IDCRT
Routines scheduled:	None
Mailboxes used:	None
Events referenced:	EVEXPL, EVKEY, EVTXT
Input arguments:	INDEX : pointer into ERXFI
Output arguments:	None
Local variables:	TIMAR : time BUFF : for ERXFI text BUFF2 : for VOTEXT phrases
Files created/changed:	STUFE, ERXFI
Files referenced:	None
Notes:	None

Title:	RPCLOK
Source file:	RPCLOK.SR
Description:	This increments 100msec clock and 500msec clock during replay and XMTs the time to BXRPL and BXACT. Then it asks for rescheduling upon exit.
Classification:	User clock
Period:	100ms
Language:	A
Activated/called by:	REPLAY, ERRHAN
Cancelled by:	REPLAY, ERRHAN
Activates/calls:	None
Cancels:	None
IPB ID's used:	None
Routines scheduled:	None
Mailboxes used:	BXACT, BXRPL
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	RPINITAC
Source file:	RPINITAC.FR
Description:	This initializes the display at the beginning of replay. It is passed a 16 word record from RPLACT or RPPACT with a one in the first word. The remaining words have display information.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	ACTIVITY
Cancelled by:	N/A
Activates/calls:	IPBOUT1
Cancels:	None
IPB ID's used:	IDIMAGES, IDSERVO
Routines scheduled:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	RECORD : 16 words beginning with a 1 or -1
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	Pictures are started so this routine can be used for P-run replay.

Title:	GETBUFF
Source file:	GETBUFF.FR
Description:	This routine is called by RDACT and SUSHAN when they reach the end of a buffer. It changes buffers for them.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RDACT, SUSHAN
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	IER : error return from read on NCRPAT
Files created/changed:	None
Files referenced:	RPLACT, RPPACT
Notes:	None

Title:	RDACT
Source file:	RDACT.FR
Description:	This fills words in SPACT.CO from a buffer in SPACT.CO for use by PMS. It is called during Phase 2.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PZEC, RZEC
Cancelled by:	N/A
Activates/calls:	SUSHAN, GETBUFF
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	IER : error return word
Local variables:	ITSB* : temporaries for SABUF elements
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SLURP
Source file:	SLURP.FR
Description:	This routine fills SPACT.CO during replay.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SURPLY
Cancelled by:	None
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SURPLY
Source file:	SURPLY.FR
Description:	This routine fills SPACT.CO for REXPLAIN at the appropriate time.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	ACTIVITY
Cancelled by:	N/A
Activates/calls:	SLURP
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SUSHAN
Source file:	SUSHAN.FR
Description:	This handles SUS records for RDACT. That is, it puts them in the correct words in SPACT.CO.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RDACT
Cancelled by:	N/A
Activates/calls:	GETBUFF
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	IER : error return agrument
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

VOICE DATA COLLECTION, SPEECH RECOGNITION, AND SPEECH UNDERSTANDING

Title:	ACTSUS
Source file:	ACTSUS.FR
Description:	This appends an environmental buffer to the speech buffer for output to the student activity file. Also, the SUS buffer is reformatted to account for "correction" and "over".
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SUS, SUSWRITE, SUSEND, RSB, SUCPH
Cancelled by:	N/A
Activates/calls:	ISABUF, ACTOUT, RNGSCHD
IPB ID's used:	None
Routines scheduled:	SUGYRO, MARKIT
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	IDUM1 : dummy array indicator IDUM2 : dummy array size IBUF1 : speech buffer to be output
Output arguments:	None
Local variables:	ICOR : correction flag IBUF2 : buffer to be appended
Files created/changed:	None
Files referenced:	None
Notes:	This schedules SUGYRO for execution by RNGCAL .5 miles hence. It also schedules MARKIT to submit a mile record .75 miles hence.

Title:	BEGIN
Source file:	BEGIN.FR
Description:	Preparations for the voice validation mode are made by this routine.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	VDC2VAL, VDC1VAL
Cancelled by:	N/A
Activates/calls:	VSRRC, VRPLD, VVUCL (user clock)
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	IER : error argument I : loop index
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	COLLECT
Source file:	COLLECT.FR
Description:	This routine collects input feature patterns from trainee voice inputs. Automatic prompts are provided only if a bad input is detected. The user must provide the initial prompt.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SPEECH
Cancelled by:	N/A
Activates/calls:	VIPON, VGIFP, VIPOFF, PRESENT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	IER : error I : loop index INP : phrase count IDNUM : phrase number ILNG : VRP length ITIM : # of time slots IREC : ptr. to IPB storage record IFPS : IFP array LTOUT : first timeout flag
Files created/changed:	FNIFP
Files referenced:	None
Notes:	None

Title:	COMBO
Source file:	COMBO.FR
Description:	This is the routine which finds the best combination of digits for SHEAD (the SUS heading routine).
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SHEAD
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routine scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	ICOMB : combinations array element 1-number of possibilities rest-numbers themselves.
Output arguments:	None
Local variables:	I : locp index TMP1, TMP2, TMP3 : temporaries
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	FILL
Source file:	FILL.FR
Description:	This routine helps fill the speech understood buffers. It determines which buffer is free and fills it from the buffer of the phrase which got the highest score from SUS.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SUS, SUCPH
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routine scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	ITYP : phrase group category
Output arguments:	None
Local variables:	I : loop index
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	FORMIT
Source file:	FORMIT.FR
Description:	This forms voice reference patterns for the requested phrase by calling SFORMIT.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SPEECH
Cancelled by:	N/A
Activates/calls:	SFORMIT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	FRZOT
Source file:	FRZOT.FR
Description:	This routine provides CRT prompts for voice data collection and validation.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PRESENT
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	IFRZ : the phrase to be prompted
Output arguments:	None
Local variables:	I : loop index J : loop index KREC : pointer into prompt file IER : error code TTOBUF : prompt buffer
Files created/changed:	None
Files referenced:	SPK.VO
Notes:	FRZOT adds "(PAUSE)" after each phrase.

Title:	HEARSAY
Source file:	HEARSAY.FR
Description:	This recognition task partitions the vocabulary by using the controller messages it receives from ISAY to select a set of resolution masks which single out the phrases the trainee is most likely to use at the time.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	LOOKOUT
Cancelled by:	Self
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	IPHZ : phase of flight IGPP : glidepath position mask IGPT : glidepath trend mask ICRP : course position mask ICRT : course trend mask IRNG : range mask IEMERG : emergency mask IOTHR : other messages mask
Output arguments:	IPHZ : set to -1 when processing is complete
Local variables:	I : loop index
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	HSCIN
Source file:	HSCDR.SR
Description:	This routine attaches the high speed correlator to the RDOS interrupt structure.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	SUSON, VDCON
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	HSCOUT
Source file:	HSCDR.SR
Description:	This routine removes the high speed correlator from the RDOS interrupt structure.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	SUSOFF, VDCOFF
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

AD-A087 190 LOGICON INC SAN DIEGO CA TACTICAL AND TRAINING SYSTE--ETC F/G 17/9
GROUND CONTROLLED APPROACH CONTROLLER TRAINING SYSTEM (GCA-CTS)--ETC(U)
JUN 80 G BARBER, J BOLLENBACHER, D BREWTON N61339-77-C-0162

UNCLASSIFIED NAVTRAEQUIPC-77-C-0162-3 NL

4 OF 8

AD
A087190

Title:	ISABUF
Source file:	ISABUF.FR
Description:	This writes to and from the environmental buffer array which contains information to be appended to SUS records.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	ISAY, ACTSUS
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	IACT : action code = 1 : retrieve buffer for ACTSUS = 2 : add buffer to queue = 3 : retrieve most recent buffer ITIME : time in 100 ms ticks = LP4 time, if IACT = 1 = CL100, if IACT = 2 IBUF : (SUSEG-1) word buffer = buffer to be added if IACT = 2
Output arguments:	IBUF : retrieved buffer, if IACT = 1,3
Local variables:	I : loop index INDEX : SUBUF index
Files created/changed:	None
Files referenced:	None
Notes:	This references common block SUSAY.CO for queue structure.

Title:	ISAY
Source file:	ISAY.FR
Description:	This is a speech understanding task which ships model controller generated final controller message number IDs to CPU 2 speech recognition modules upon initiation of student voice input. These messages identify a valid set of final controller phrases. The same information is sent out to the activity replay file. Aircraft position parameters of interest to SUS and PMS are also stored.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	PB23SUB
Cancelled by:	SUSTRM
Activates/calls:	IPBOUT1, ISABUF
IPB ID's used:	IDSPEECH, IDHEARSAY
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVPHZ, EVVST, EVVIN
Input arguments:	None
Output arguments:	None
Local variables:	I : loop index NBUF : buffer array of words to be put on the queue ITRND : trend value of 1 IBUF : array 7, model message set
Files created/changed:	None
Files referenced:	None
Notes:	ISAY's overlay is released by SUSTRM.

Title:	LEVEL1
Source file:	LEVEL1.FR
Description:	This routine is activated when the end of student input is detected. It ships the input level across the IPB to voice data collection.
Classification:	Task
Period:	None
Language:	F
Activated called by:	TASKOUT
Cancelled by:	IPBIN1 when directed by VDCOFF
Activates/calls:	IPBOUT1, PANIT
IPB ID's used:	IDLEVEL
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVVIN
Input arguments:	None
Output arguments:	None
Local variables:	IARG : speech input level
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	LEVEL
Source file:	LEVEL.FR
Description:	This routine receives the input speech level from CPU 1, sets it for the CPU 2 speech routines, and generates an end of speech input wakeup.
Classification:	Subroutine
Period:	None
Language:	F
Activated called by:	LOOKOUT
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	
Routines scheduled:	
Cancels:	None
Mailboxes used:	None
Events referenced:	EVVIN
Input arguments:	ILVL : speech level as measured by trainee panel
Output arguments:	ILVL : set to -1 when level has been copied
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PRESENT
Source file:	PRESENT.FR
Description:	This presents phrase prompts on the requested prompt device.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	COLLECT, VDC2VAL, TEST, IPBIN2, INIT2RT, TALKOUT
Cancelled by:	N/A
Activates/calls:	FRZOT, IPBOUT2
IPB ID's used:	IDVSPRES
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	IPDEV : requested prompt device 0 = IPB CRT prompt 1 = $VRO 2 = $TTO 3 = audio IPLST : list of phrases to be prompted (array 6)
Output arguments:	IPDEV : set to -1 if IPB call
Local variables:	INDX : phrase count index
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	RSB
Source file:	RSB.FR
Description:	This handles the release of the speech buffer for SUS. That is, it resets all of the working buffers, sends off the buffer (to PMS in Phase 2 or the replay file in Phase 3) if it is ready. If the buffer is not ready (for example, looking for correction or over) SUSWRITE is tasked to write it out in one second if no further phrases are heard.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SUS
Cancelled by:	N/A
Activates/calls:	SUSWRITE, ACTSUS, TSKERRDLY
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	I, J : loop control IER : error argument
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SAID
Source file:	SAID.FR
Description:	This is a speech recognition task which ships recognitions over to CPU 1.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	SUSON
Cancelled by:	SUSOFF
Activates/calls:	IPBOUT2
IPB ID's used:	IDSUS
Routines scheduled:	None
Cancels:	None
Mailboxes used:	BXCOG
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	IMSG : message from recognition
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SDIGIT
Source file:	SDIGIT.FR
Description:	This routine processes digits for speech understanding. It also checks for possible "miles from touchdown" advisory. If it finds that a digit was recognized by the speech recognizer then the digit is stored and a new value is computed for the current digit-string. If a digit was not one of the choices, it checks if any other phrase is expecting a digit. If so, it signals that the digits are through and calculates the value for the digits seen. If no one is expecting digits, SDGIT checks to see if this message could be a correct "X miles from touch down". If so it builds such a message in its work buffer.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SUS
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	J, K : loop indices TMP : combination count IFLS : digit choice count flag TMP1 : temporary
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SFORMIT
Source file:	SFORMIT.FR
Description:	This forms voice reference patterns for the requested phrases.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	FORMIT
Cancelled by:	N/A
Activates/calls:	VGVRP
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	NREP : # of repetitions necessary IER : error IDNUM : phrase # INTS : # of VRP slots to be stored NRECS : # of IFP records to be read IFPD : IFP working array IVRPR : VRP storage array IPT : loop index ptr. to phrase
Files created/changed:	FNVRP
Files referenced:	FNIFP
Notes:	None

Title:	SHEAD
Source file:	SHEAD.FR
Description:	This processes heading messages for speech understanding.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SUS
Cancelled by:	N/A
Activates/calls:	COMBO
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	J : loop index TMP : temporary NUM : heading combinations array
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SMISH
Source file:	SMISH.FR
Description:	This processes missed approach recognitions for speech understanding. If the speech recognizer has not recognized a part of the missed approach message then SMISH looks to see if it was expecting a recognition. If so it clears out its working buffer and sets a flag indicating that the missed approach message was out of order. Then SMISH quits. If the speech recognizer has recognized part of a missed approach message then SMISH concatenates this new recognition to the existent phrase.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SUS
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	I : loop index TMP1, TMP2 : temporaries
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SMOTHR
Source file:	SMOTHR.FR
Description:	This routine looks out for low confidence phrases and tries to choose the best one based on a priori message information.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SUS
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	K : loop index IREF : reference index to next syntactical phrase IND : stores above index ISIGN : call sign number
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SMREC
Source file:	SMREC.FR
Description:	This processes unrecognizable phrases. If a recognized phrase is followed by 3 digits which match the present call sign digits, then it is assumed that the misrecognized phrase was the call sign.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SUS
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	I,K : loop index ISIGN : call sign digits
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SPEECH
Source file:	SPEECH.FR
Description:	SPEECH initiates the speech related routines which accomplish voice pattern collection, voice pattern validation, recognition, and understanding.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	INIT2RT, LOOKOUT
Cancelled by:	Self
Activates/calls:	IPBOUT2, VDCON, COLLECT, FORMIT, VDC1VAL, VDC2VAL, VDCOFF, SUSON, SUSOFF
IPB ID's used:	IDAWAKE
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVPHZ
Input arguments:	IBRNCH : command code 1 = start VDC 2 = collect IFP 3 = form VRP 4 = validate 5 = validate, no prompt 6 = stop VDR 7 = start SUS 8 = stop SUS SELST(1) : % val for command 4 or phrase # SELST(2) - SELST(7) : phase #'s SEDEV : speech device
Output arguments:	IBRANCH : set to -1
Local variables:	I : loop index
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SPINIT
Source file:	SPINIT.FR
Description:	Initializes trainee-independent speech data.
Classification:	Subroutine
Period:	N/A
Language:	F
Activated/called by:	TUNIT
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	IER : error code
Files created/changed:	None
Files referenced:	SPK.VO
Notes:	None

Title:	STIFLE
Source file:	STIFLE.FR
Description:	This waits for STIFLE key input. The student selects the stop voice test key when he tires of the validation mode (or feels frustrated or antagonistic).
Classification:	Task
Period:	None
Language:	F
Activated/called by:	LOOKOUT
Cancelled by:	Self
Activates/calls:	VIPOFF
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	BXCOG
Events referenced:	None
Input arguments:	IDUMMY : argument for IPB release
Output arguments:	None
Local variables:	IMSG : STIFLE message
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SUCOVFLG
Source file:	SUCOVFLG.FR
Description:	This sets correction and over flags in replay buffer for SUS.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SUS
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	SPKCORRECTION : T if correction spoken SPKOVER : T if over spoken
Output arguments:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SUCPH
Source file:	SUCPH.FR
Description:	This checks out the phrase for SUS and releases it if there are no conflicts.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SUS
Cancelled by:	N/A
Activates/calls:	FILL, ACTSUS, SUSWRITE, TSKERRDLY
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	MIKE : mike key status HIGH : phrase with highest concatenation count
Output arguments:	HIGH : -1 if SUS is to skip call to RSB
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SUGYRO
Source file:	SUGYRO.FR
Description:	This routine writes a special record after speech understanding recognizes a "Heading..." message given under no gyro conditions.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RNGCAL (by ACTSUS)
Cancelled by:	N/A
Activates/calls:	ACTOUT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SUS
Source file:	SUS.FR
Description:	This is the speech understanding controller. It processes speech recognition inputs which are shipped to it. It does this by calling various routines that process different types of phrases. When a routine has found a complete message, SUS sends that message to APE, MODEL, and PMS.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	TASKOUT
Cancelled by:	Self
Activates/calls:	SHEAD, SWIND, SMISH, SDIGIT, SMREC, SMOTHR, FILL, ACTSUS, RSB, SUCPH, SUCOVFLG
IPB ID's used:	None
Routines scheduled:	None
Cancels:	SUSWRITE
Mailboxes used:	None
Events referenced:	None
Input arguments:	IHSEC : LP4 half second time IMSEC : LP4 100 msec time IREC1 : first choice phrase recognized IHDG : heading flag IWND : wind flag IREC2 : second choice phrases IMAP : missed approach flag
Output arguments:	IHSEC : set to -1 for IPB
Local variables:	I,J : loop indices LCOROVR : correction/over input flag HIGH : phrase concatenation group winner MIKE : mike keyed/unkeyed SPKOVER : T, if "over" SPKCORRECTION : T, if "correction"
Files created/changed:	None
Files referenced:	None
Notes:	Initialize all buffers and flags to -1 upon program load! However, SSUSE initially should be 1, SUS shall use relative #'s only

Title:	SUSEND
Source file:	SUSEND.FR
Description:	SUSEND is called to clean out the SUS buffers and to output any remaining information.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SUSTRM
Cancelled by:	N/A
Activates/calls:	ACTSUS, PMCLR
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	I : loop index
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SUSOFF
Source file:	SUSOFF.FR
Description:	This turns off all speech understanding modules.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SPEECH
Cancelled by:	N/A
Activates/calls:	VIPOFF, HSCOUT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	SAID, VSRRC
Mailboxes used:	BXREC
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	IER : error argument
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SUSON
Source file:	SUSON.FR
Description:	SUSON activates and prepares all speech understanding files and tasks.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SPEECH
Cancelled by:	N/A
Activates/calls:	VRPLD, VSRRC, SAID, VIPON, HSCIN
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVVRPD
Input arguments:	None
Output arguments:	None
Local variables:	IPCT : validation % IER : error I : loop index
Files created/changed:	None
Files referenced:	VRP.VO
Notes:	None

Title:	SUSTRM
Source file:	SUSTRM.FR
Description:	This is the SUS termination routine. It stops speech recognition by killing ISAY and SUSWRITE and then calling SUSEND.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	P2FRZ, P2RNSTOP, RUNSTOP
Cancelled by:	N/A
Activates/calls:	IPBOUT1, SUSEND
IPB ID's used:	IDSPEECH
Routines scheduled:	None
Cancels:	ISAY, SUSWRITE
Mailboxes used:	None
Events referenced:	EVPHZ
Input arguments:	None
Output arguments:	None
Local variables:	IER : error argument STATUS : status argument to pass to IDST
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SUSWRITE
Source file:	SUSWRITE.FR
Description:	This writes out a speech buffer for SUS whenever a timeout occurs.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	RSB, SUCPH
Cancelled by:	SUS, SUSTRM
Activates/calls:	ACTSUS
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SWIND
Source file:	SWIND.FR
Description:	This routine processes wind messages for speech understanding. If the speech recognizer has not recognized a part of a wind message, SWIND clears out its buffer in SHUSH.CO and if SWIND was in the middle of recognizing a wind message, it sets a flag indicating that the phrase was not in the correct order. It then sets its concatenation number to -1 and quits. If the speech recognizer has recognized a part of the wind message, SWIND concatenates this new part to the existent phrase.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SUS
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	I,J : loop indices IWHDG : integer wind hdg. IWSP : integer wind speed TMP, TMPA, TMP1, TMP2 : temporaries
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	TERMINATE
Source file:	TERMINATE.FR
Description:	This cleans up for validation mode.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	VDC1VAL, VDC2VAL
Cancelled by:	N/A
Activates/calls:	VIPOFF
IPB ID's used:	None
Routines scheduled:	None
Cancels:	VVUCL, VSRRC
Mailboxes used:	BXREC
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	IER : error argument
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	TEST
Source file:	TEST.FR
Description:	This performs a validation run for the given phrase(s). The run is terminated by the attainment of the requested validation percentage in three consecutive repeats or by student request.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	VDC1VAL
Cancelled by:	N/A
Activates/calls:	VIPON, VIPOFF, PRESENT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	BXCOG
Events referenced:	EVKEY
Input arguments:	None
Output arguments:	None
Local variables:	LTOUT : long timeout flag IER : error argument INDX : phrase counter I,J : loop indices MISS : mistake count NVLID : number of validated input rounds IMSG : mail from voice recognition, TIMEOUT, STIFLE TSTBUF : recognition holding buffer
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	VALYZ
Source file:	VALYZ.SR
Description:	This routine analyzes the recognition choice(s). If only one choice was made, no further analysis is done and a high confidence factor is assigned to the recognition. If there are two choices, the Breaux test is performed.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	VOVEX
Cancelled by:	N/A
Activates/calls:	VMAP, VCORR
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	ARG1 : index of best choice ARG0 : level of confidence = -1, high confidence, or = second choice index, low conf.
Local variables:	ANLCT : loop counter for Breaux test ANLT1 : ptr. to VRP 1 ANLT2 : ptr. to VRP 2 ANLX1 : first score ANLX2 : second score TMP1 : VRP 1 block pointer; score TMP2 : VRP 2 block pointer; score TMP3 : IFP normalization counter VAI5 : IFP ptr. VAI6 : VRP 1 ptr. VAI7 : VRP 2 ptr. ASUM : score sum CNT1 : choice table loop count 1 CNT2 : choice table loop count 2
Files created/changed:	None
Files referenced:	None
Notes:	Assemble this routine with VSIFPHDR.SR for common variable macro definitions. Program constants: K.25K=256. : VRP buffer area offset from score table start K50.=50. : normalization multiplication constant C2=2 : choice table entry length

Title:	VCHOS
Source file:	VCHOS.SR
Description:	The highest scores are chosen from among the scores computed. A minimum score must be met. A second choice is reported if it exceeds the minimum score and is close to the winner.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	VOVEX
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	ARG0 : 0 = matches found 1 = no matches found
Local variables:	CHCNT : counter WCHTA : working choice table address CHOT2 : choice table copy buffer VFLGB : closeness flag SELCT : counter for select routine MAX : high score
Files created/changed:	None
Files referenced:	None
Notes:	Assemble this routine with VSIFPHDR. Constants: KM1=1 : no close score flag K200=200 : score area length

Title:	VCOMP
Source file:	VCOMP.SR
Description:	This routine performs the comparison between the IFP and flagged VRPS.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	VOVEX
Cancelled by:	N/A
Activates/calls:	VCORR, VMAP
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	ARG0 : number of VRPS to be investigated
Output arguments:	None
Local variables:	CITEM : pointer to locator CNTR : count down counter CSCOR : pointer to score location COMX : score VAI5 : VRP ptr. VAI6 : IFP ptr. COMCT : slot counter SUM : running correlation sum
Files created/changed:	None
Files referenced:	None
Notes:	Assemble this routine with VSIFPHDR.SR. Constants: K50.=50. : normalization multiplier

Title:	VCORR
Source file:	VCORR.SR
Description:	This routine computes the correlation between two feature patterns using the high speed correlator.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	VCOMP, VALYZ
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	VDC1VAL
Source file:	VDC1VAL.FR
Description:	This validates the given phrases to the requested percentage accuracy. Prompts are reissued until the percentage meets the requirement.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SPEECH
Cancelled by:	N/A
Activates/calls:	BEGIN, TEST, TERMINATE
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	I,J : loop indices
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	VDC2VAL
Source file:	VDC2VAL.FR
Description:	This validates student voice inputs for VRP defined phrases. Recognized phrases are mimicked after the student input level drops.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SPEECH
Cancelled by:	Self
Activates/calls:	BEGIN, VIPON, VIPOFF, PRESENT, TERMINATE
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	BXCOG
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	I : loop index IER : error argument IMSG : mail from STIFLE, TIMEOUT, recognition IFRZ : input phrase PRESENT array LTOUT : logical first timeout flag
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	VDCOFF
Source file:	VDCOFF.FR
Description:	This turns voice data collection and validation off.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SPEECH
Cancelled by:	N/A
Activates/calls:	IPBOUT2, HSCOUT
IPB ID's used:	IDKILL, IDLEVEL1
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	IER : error argument I : loop index
Files created/changed:	IFP.VO, VRP.VO
Files referenced:	None
Notes:	None

Title:	VDCON
Source file:	VDCON.FR
Description:	This routine reads all the files necessary for VDC and validation and initializes all necessary trainee dependent arrays.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SPEECH
Cancelled by:	N/A
Activates/calls:	IPBOUT2, HSCIN
IPB ID's used:	IDLEVEL1
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	IER : error
Files created/changed:	None
Files referenced:	IFP.VO, VRP.VO
Notes:	None

Title:	VGIFP
Source file:	VGIFP.FR
Description:	This creates input feature patterns (IFPs) for voiced inputs. If an input is not received within 20 seconds, a message is returned to the caller.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	COLLECT
Cancelled by:	N/A
Activates/calls:	VUCLK, VIFP
IPB ID's used:	None
Routines scheduled:	None
Cancels:	VUCLK
Mailboxes used:	BXREC
Events referenced:	None
Input arguments:	None
Output arguments:	IFP : created IFP IVTIM : number of 2 ms periods which the IFP spans IVTS : number of time slots stored for VRP
Local variables:	IER : error
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	VGVRP
Source file:	VGVRP.FR
Description:	This forms a voice reference pattern using the input feature patterns given.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SFORMIT
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	IFPS : array if IFPs previously collected NREP : number of repetitions IVTS : number of time slots for VRP
Output arguments:	VRP : VRP array
Local variables:	I,J,K : loop indices ISUM : # of times feature bit set
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	VIFP
Source file:	VIFP.SR
Description:	This routine creates an input feature pattern from raw data.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	VGIFP
Cancelled by:	N/A
Activates/calls:	VICOM, VSIFP
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	IFP : location of IFP storage IVTIM : location of input time storage IVTS : number of time slots
Output arguments:	IFP : new IFP IVTIM : input time
Local variables:	VAI3, VAI4, VAI5 : counters VTSL : time slot loop count REM : remainder time slots STEP : number of time slots per group BCTR : bit counter LIM : beginning of next group WSCT : words per slot count
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	VIPOFF
Source file:	VIPDR.SR
Description:	This routine disables the TTI 500.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	COLLECT, STIFLE, SUSOFF, TERMINATE, TEST, VDC2VAL, SKPRO
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	VIPON
Source file:	VIPDR.SR
Description:	This informs RDOS of the TTI 500 (the THRESHOLD device).
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	COLLECT, SUSON, TEST, VDC2VAL, SKPRO
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	VMAP
Source file:	VMAP.SR
Description:	VMAP attempts to locate the VRP in core. It remaps if necessary, and establishes the pointers to the IFP and normalization factor buffer. It sets the number of reference time slots.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	VCOMP, VALYZ
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	VOICTST
Source file:	VOICTST.FR
Description:	This routine initiates the voice validation mode for the trainee as a result of a keyboard request.
Classification:	Task
Period:	None
Language:	F
Activated called by:	PZREQ
Cancelled by:	None
Activates/calls:	IPBOUT1, MENU, LOGVT
IPB ID's used:	IDSPEECH, IDSTIFLE
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVPHZ, EVSTP
Input arguments:	None
Output arguments:	None
Local variables:	None
Files referenced:	None
Notes:	None

Title:	VOVEX
Source file:	VOVEX.SR
Description:	VOVEX finds likely candidates for recognition, then compares and scores these VRPS for the given IFP.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	VSRRC
Cancelled by:	N/A
Activates/calls:	VRPRT, VCOMP, VCHOS, VALYZ, VSPCL
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	PSCNT : pass count CNTR : match counter TMP1 : AC0 argument for VSPCL call TMP2 : AC1 argument for VSPCL call ARET : return address
Files created/changed:	None
Files referenced:	None
Notes:	Assemble this routine with VSIFPHDR.SR. Program constants: K16.=16. : 16 time slots K32.=32. : 32 time slots K3=3 : block move constant K200=200 : score offset K40=40 : 16 time slot offset

Title:	VRPLD
Source file:	VRPLD.FR
Description:	This loads voice reference patterns into virtual memory.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	BEGIN, SUSON
Cancelled by:	Self
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVVRPD
Input arguments:	None
Output arguments:	None
Local variables:	IER : error argument
Files created/changed:	None
Files referenced:	VRP.VO
Notes:	None

Title:	VRPRT
Source file:	VRPRT.FR
Description:	Any VRP whose identification tag matches a model controller selection for the bits set in the associated resolution mask, is flagged with its location pointer. VRPRT returns a count of phrases flagged.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	VOVEX
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	IPASS : pass number through recognition cycle for present input ITABLE : locator table
Output arguments:	IPASS : set to 3 if possibilities exhausted in this pass ITABLE : locators for selected VRPS MANY : count of phrases flagged
Local variables:	INDEX : mask index, # masks to be used I,J : loop indices ITMP : temporary TSBIT : phase bit # USEMSK : mask array to be used USERES : resolution array to be used
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	VSPCL
Source file:	VSPCL.FR
Description:	This routine performs vocabulary specific proc.ssing. Special message types which incorporate variable numerical phrases are identified and masks are prepared for their reception.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	VOVEX, SRXMT
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	BXCOG
Events referenced:	None
Input arguments:	ICHZ2 : confidence indication (second choice item or -1) ICHZ1 : chosen item index
Output arguments:	None
Local variables:	MSG : message for mailbox BXCOG = 1, no item = 2, too short = 3, too long = 5, good item
Files created/changed:	None
Files referenced:	None
Notes:	The arguments passed are relative indices which range from 0 to NVRP-1 or -1, -2, -3.

Title:	VSPRES
Source file:	VSPRES.FR
Description:	This routine presents automated voice prompt for CPU 2 speech prompt or for recognition presentation. Pauses are added between phrases to accentuate the stylization.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	P1PRM, TASKOUT
Cancelled by:	Self
Activates/calls:	GLIB
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	ISRC : 0, request source is PRESENT (via IPB) : 1, source is P1PRM IDEV : GLIB output device IRAY : phrase array (6)
Output arguments:	ISRC : -1, if IPB call
Local variables:	INDEX : filler array index and GLIB source ARRAY : new phrase array (13) I : loop index
Files created/changed:	None
Files referenced:	None
Notes:	A pause is not added after the digitized voice phrases.

Title:	VSRRC
Source file:	VSRRC.FR
Description:	This is a voiced speech recognition task which awaits student inputs to be processed for recognition.
Classification:	Task
Period:	None
Language:	A
Activated/called by:	BEGIN, SUSON
Cancelled by:	SUSOFF, TERMINATE
Activates/calls:	VOVEX
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	ARG1 : validation %, or 0 (for default score)
Output arguments:	None
Local variables:	.STASH : ptr. to storage area for IFP .BUFFER : ptr. to buffer BUFFER : stash buffer for short IFP VAI3 : buffer 1 ptr. VAI4 : buffer 2 ptr. VAI5 : VSCAR ptr. VTSL : loop count REM : remainder samples STEP : # samples/slot BCTR : bit counter LIM : step limit WSCT : words/slot count
Files created/changed:	None
Files referenced:	None
Notes:	Assemble VSIFPHDR, VICOMBDR, MAIL2HDR with this routine Program constants: MARR : 115-ASCII "M" for map error K6 : 6 VIPDR comm. packet size K16 : 16 timeslots K32 : 32 timeslots K1K : 1K words M31 : VSCAR clearance count GD16 : 16 bits/word VSIFP parameters: KWIN VSNWBLK MDFLT VSMDFLT

Title:	VUCLK
Source file:	VUCLK.SR
Description:	Speech data collection user clock.
Classification:	User clock
Period:	6 seconds (ticks only once)
Language:	A
Activated/called by:	VGIFP
Cancelled by:	VGIFP
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	BXREC
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	VVUCL
Source file:	VVUCL.SR
Description:	Validation timeout clock.
Classification:	User clock
Period:	100 msec
Language:	A
Activated/called by:	BEGIN
Cancelled by:	TERMINATE
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	BXCOG
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

AIRCRAFT/PILOT/ENVIRONMENT SIMULATION

Title:	APE1NIT
Source file:	APE1NIT.FR
Description:	This is the APE subsystem initializer/invoker part 1 (for APEX only).
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	APENIT, PZDEMO
Cancelled by:	None
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	I : loop control DYUM : multiplication factor for pilot type
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	APE2NIT
Source file:	APE2NIT.FR
Description:	This is the APE subsystem initializer/invoker part 2 (for APEX only).
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	APENIT
Cancelled by:	None
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	SCOYD(4) : standard climbout rate of climb by A/C type SFAAS(4) : standard final airspeed by A/C type SCOAS(4) : standard climbout airspeed by A/C type FNOCOPY(5) : pilot random advisory disregard percentages SPTAS(4) : standard pilot airspeed by A/C type BYDMAX : upper boundary for climb rate BYDMIN : lower boundary for climb rate BHDMAX : upper boundary for turn rate BHDMIN : lower boundary for turn rate
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	APE3NIT
Source file:	APE3NIT.FR
Description:	This is the APE subsystem initializer/invoker part 3 (for APEX only).
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	APENIT
Cancelled by:	None
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	SARVYDI : temp for initialization SYDI : temp for initialization SA1HD : temp for initialization SA1YD : temp for initialization SA1AS : temp for initialization
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	APE4NIT
Source file:	APE4NIT.FR
Description:	This is the APE subsystem initializer/invoker part 4 (for APEX only).
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	APENIT
Cancelled by:	None
Activates/calls:	APEX
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	I : loop control TEMP : temporary
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	APE5NIT
Source file:	APE5NIT.FR
Description:	This is the APE subsystem initializer/invoker part 5 (for APRAX, APREX only).
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	APENIT
Cancelled by:	N/A
Activates/calls:	APRAX, APREX
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	APENIT
Source file:	APENIT.FR
Description:	This is the APE subsystem initialization executive.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	P23SUB, P1AC
Cancelled by:	N/A
Activates/calls:	IPBOUT1, APEINIT, APE2NIT, APE3NIT, APE4NIT, APE5NIT
IPB ID's used:	IDIMAGES
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	APEX
Source file:	APEX.FR
Description:	This simulates a pilot flying the aircraft under user-specified wind conditions and responding both verbally and by control movement to advisories transmitted to him by the GCA controller.
Classification:	Subroutine
Period:	0.5 seconds
Language:	F
Activated/called by:	APE4NIT
Cancelled by:	N/A
Activates/calls:	WIND, THINKPILOT, SPEAKPILOT, MOVEPILOT, RADAR
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	BXCYC, BXRZ
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	IDUMMY : to pass to REC
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	APRAX
Source file:	APRAX.FR
Description:	This simulates an aircraft flying precisely on glide-slope but oscillating between two user-specified "boundary" course zones in the x-z (course) plane.
Classification:	Task
Period:	0.5 sec
Language:	F
Activated/called by:	APE5NIT
Cancelled by:	N/A
Activates/calls:	RADAR
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	BXCYC, BXRZ
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	FMAX : X coordinate of right "boundary" for Z=ACZ FACTOR : X displacement of target center from CRS as % of BLIPSIZ BLIPSIZ : real space height of displayed target I : loop control IDUMMY : used in REC argument list FMIN : X coordinate of left "boundary" for Z=ACZ
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	APREX
Source file:	APREX.FR
Description:	This simulates an aircraft flying precisely on course but oscillating between two user-specified "boundary" course zones in the y-z (elevation) plane.
Classification:	Task
Period:	0.5 seconds
Language:	F
Activated/called by:	APE5NIT
Cancelled by:	Self
Activates/calls:	RADAR
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	BXCYC, BXRZ
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	FMIN : Y-coordinate of lower "boundary" for Z=ACZ FMAX : Y-coordinate of upper "boundary" for Z=ACZ I : loop control IDUMMY : to pass to REC IER : error argument BLIPSIZ : real-space size of displayed target FACTOR : Y displacement of target from glidepath as % of blipsize GP : Y-coordinate of glidepath for Z=ACZ
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	CLOK
Source file:	CLOK.SR
Description:	This is the CPU 1 user clock routine. It increments CL100 every 100 msec and CLTICK every .5 seconds (range 0-end of run).
Classification:	User clock routine
Period:	100 msec
Language:	A
Activated/called by:	KTEACH, P1AC, P23SUB, PHZ1
Cancelled by:	P2FRZ, RUNKILL, KTEACH
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	BXCYC
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	CLOK2
Source file:	CLOK2.SR
Description:	This is the CPU 2 user clock routine.
Classification:	User clock
Period:	100 msec
Language:	A
Activated/called by:	IPBIN2
Cancelled by:	SKPRO
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	CLOKF
Source file:	CLOKF.SR
Description:	This is the CPU 2 foreground user clock.
Classification:	User clock
Period:	100 msec
Language:	A
Activated/called by:	STARTF
Cancelled by:	STARTF
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	BXTIM
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	CONCEIVETH
Source file:	CONCEIVETH.FR
Description:	This simulates a pilot deciding how to maneuver his aircraft in response to a new advisory.
Classification:	Subroutine
Period:	called from 0 to N times each 0.5 seconds, where N is the number of SUS buffers.
Language:	F
Activated/called by:	THINKPILOT
Cancelled by:	N/A
Activates/calls:	TIMSCHD
IPB ID's used:	None
Routines scheduled:	IMOFF
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	ADVID : phrase id number of advisory just "copied" ADVHDG : heading accompanying advisory, if any (degrees)
Output arguments:	None
Local variables:	EVARDELY : variance associatd with EDELY EVARYDI : variance associated with EYDI GPZONE : G/P zone corresponding to advisory, if any ALPHA : Kalman filter coefficient GAMMA : Kalman filter coefficient (1.0-ALPHA) BLIPHEIGHT - real space height of target mark EDELY : current pilot estimation of height above G/P EVDI : latest pilot estimation of ideal R.O.D.
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	DEDUCETHEC
Source file:	DEDUCETHEC.FR
Description:	This simulates the pilot deciding what to reply upon receiving a new advisory.
Classification:	Subroutine
Period:	called from 0 to N times each 0.5 seconds, where N is the number of SUS buffers.
Language:	F
Activated/called by:	THINKPILOT
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	ADVID : phrase id number of advisory just "copied" ADVHDG : heading accompanying advisory, if any
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	GYROKILL
Source file:	GYROKILL.FR
Description:	GYROKILL kills the aircraft's gyro to simulate an in-flight gyro failure.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RNGCAL (by P23SUB)
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	IMOFF
Source file:	IMOFF.FR
Description:	This writes special record when a waveoff is to be executed.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by CONCEIVETH, LOST, OLT, PLTWAVESHI)
Cancelled by:	ENDFEED
Activates/calls:	ACTOUT, ENDFEED, TIMSCHD
IPB ID's used:	None
Routines scheduled:	None
Cancels:	FEED, SAYIT, HOLD, IMOFF
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	IER : error argument STATUS : to pass to IDST
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	MOVEPILOT
Source file:	MOVEPILOT.FR
Description:	This simulates a pilot monitoring the aircraft instruments and attempting to manipulate the flight controls so as to maintain a certain pre-conceived instrument "track". It also simulates motion of the aircraft in response to steady-state and gusting wind and pilot manipulation of flight controls.
Classification:	Subroutine
Period:	0.5 seconds
Language:	F
Activated/called by:	APEX
Cancelled by:	N/A
Activates/calls:	TIMSCHD
IPB ID's used:	None
Routines scheduled:	LOW
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	I, JK : loop control NEWAS, NEWHD, NEWXD, NEWYD, NEWZD, NEWXZS, NEWGS, NEWX, NEWY, NEWZ, NEWH, NEWHX, NEWHZ, NEWWGA, HW, HWM, BLIPSIZE, FACTOR, R(3) : temporaries
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	NEWADVISOR
Source file:	NEWADVISOR.FR
Description:	This returns "true" if the SUS buffer to which the pointer SSAPEP points contains an advisory as-yet unprocessed by APE and is marked "buffer ready"; it returns "false" otherwise. If "true", ADVID contains the advisory's phrase id number, and ADVHDG contains the advisory's heading, if any, upon return; SSAPEP is advanced to the next SUS buffer. If "false", ADVID and ADVHDG are undefined upon return, and SSAPEP is not advanced.
Classification:	Function
Period:	Called from 1 to N times each 0.5 seconds, where N is the number of SUS buffers.
Language:	F
Activated/called by:	THINKPILOT
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	ADVID : phrase id number for new advisory xmtd ADVHDG : heading for new advisory XMTed, if any
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PLTASSUMES
Source file:	PLTASSUMES.FR
Description:	This returns "true" if there was no key-unkey or ctrlr speech during the preceeding five seconds; else it returns "false".
Classification:	Function
Period:	0.5 seconds
Language:	F
Activated/called by:	THINKPILOT
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	DUMMY : dummy argument (logical function subroutines may not have empty argument lists)
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PLTCOPIEDN
Source file:	PLTCOPIEDN.FR
Description:	This routine returns "true" if pilot "copied" the new advisory which was transmitted; returns "false" if the pilot "missed" the advisory. The pilot always "copies" advisories which are neither course nor glidepath advisories; all other advisories are equally likely to be "missed", the likelihood dependent on pilot skill-level only.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	THINKPILOT
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	ADVID : advisory type identifier (phrase number)
Output arguments:	None
Local variables:	ADVID : phrase ID number of advisory ADVHDG : accompanying heading, if any (deg mag)
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PLTDECIDES
Source file:	PLTDECIDES.FR
Description:	This returns "true" if either (1) the ctrlr queried "how do you hear" one second ago and KYLVL "copied" three consecutive weak-but-clear advisories over a period of at least 1.5 seconds, the latest of which advisories the pilot copied precisely one second ago; returns "false" otherwise.
Classification:	Function
Period:	0.5 seconds
Language:	F
Activated/called by:	THINKPILOT
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	DUMMY : dummy argument (logical function subroutines may not have empty argument lists)
Output arguments:	None
Local variables:	LEVELWEAK : true if KYLVL is less than KYMINLVL
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PLTWAVESHI
Source file:	PLTWAVESHI.FR
Description:	This routine simulates pilot deciding to execute missed approach without having received an explicit waveoff from the GCA controller.
Classification:	Subroutine
Period:	None
Language:	F
Activated called by:	THINKPILOT CONCEIVETH
Cancelled by:	N/A
Activates/calls:	TIMSCHD
IPB ID's used:	None
Routines scheduled:	IMOFF
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SPEAKPILOT
Source file:	SPEAKPILOT.FR
Description:	This produces pilot verbal replies to GCA controller advisories.
Classification:	Subroutine
Period:	0.5 seconds
Language:	F
Activated/called by:	APEX
Cancelled by:	N/A
Activates/calls:	GLIB, HOWNOW
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ISRC : source identifier and error argument
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	THINKPILOT
Source file:	THINKPILOT.FR
Description:	This is the pilot thought-process module. It simulates the pilot receiving and either "copying" or "missing" GCA controller advisories. It simulates a pilot deciding "ideal" dynamic and verbal response to each advisory he has "copied". It simulates a pilot continually monitoring the controller's VX level. It simulates a pilot deciding to execute a missed approach in the event the pilot should assume that radio contact is lost.
Classification:	Subroutine
Period:	0.5 seconds
Language:	F
Activated/called by:	APEX
Cancelled by:	N/A
Activates/calls:	PLTASSUMES, PLTDECIDES, NEWADVISOR, PLTCOPIEDN, CONCEIVETH, DEDUCETHEC, PLTWAVESHI
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	NBUFCHECKD : number of SUS buffers examined so far this cycle ADVID : phrase ID number of advisory ADVHDG : accompanying heading, if any
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	WIND
Source file:	WIND.FR
Description:	This simulates the wind blowing acrcss the approach track.
Classification:	Subroutine
Period:	0.5 seconds
Language:	F
Activated/called by:	APEX
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	R : temporary
Files created/changed:	None
Files referenced:	None
Notes:	None

RADAR SIMULATION

Title:	LOOKUP
Source file:	LOOKUP.FR
Description:	This routine determines target size based on range, and clips the target to fit within display boundaries. The display boundaries vary depending on whether the servo is activated.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RADAR
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	HALFSIZE : half of correct target size SVAHISLOPE : high slope on azimuth from servo SVALOSLOPE : low slope on azimuth from servo SVEHISLOPE : high slope on elevation from servo SVELOSLOPE : low slope on elevation from servo Y1TEMP : Y point on upper azimuth sweep that corresponds to the range. Y2TEMP : Y point on lower azimuth sweep that corresponds to the range Y3TEMP : Y point on upper elevation Y4TEMP : Y point on lower elevation YAHI : temporary for change in y on hi azimuth YALO : temporary for change in y on lo azimuth YEHI : temp for change in y on hi elevation YELO : temp for change in y on lo elevation
Files created/changed:	None
Files referenced:	None
Notes:	This routine computes the slope from the target's top and bottom, and compares that slope with that of the servo display area. If the slope is too low or too high compared with the slopes of the display area, the Y located on the actual line of the display area with the same XI for range is used as the Y coordinate for the target. The parameter in LOOKUP concerns the value XPAZ (3,2) located in XPOSE.CO. The parameter must be changed if XPAZ(3,2) is changed.

Title:	RADAR
Source file:	RADAR.FR
Description:	This routine simulates the precision approach radar by converting aircraft position information into graphics display screen coordinates. It also stores these screen coordinates plus servo position, wind speed and heading for use by REPLAY.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	APEX, APRAX, APREX
Cancelled by:	N/A
Activates/calls:	LOOKUP, IPBOUT1, TIMSCHD
IPB ID's used:	IDPICUP
Routines scheduled:	LOST
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	PCMSG : 1 for normal, 2 for end of run Actually, APE calls with GZGO, so resetting PCMSG here changes that common variable.
Local variables:	XDELTA : distance in feet from radar X : receptacle for ACOFF, offset from center line in feet Y : receptacle for ACALT, altitude in feet Z : receptacle for ACRNS, range in feet YE : temporary for center of elevation blip YETEMP : same as above YA : temporary for center of azimuth blip X1STAR : distance in coordinates from order PCX1 : range in screen coordinates PCX2 : range in screen coordinates on 2nd pass PCY1 : upper part of azimuth target PCY2 : lower part of azimuth target PCY3 : upper of elevation target PCY4 : lower of elevation target PCSPHD : heading and speed of wind IRDR : array equivalenced to items for RPLDSP
Files created/changed:	None
Files referenced:	RPLDSP, RPPDSP : radar replay files
Notes:	None

DISPLAY ROUTINES

Title:	CHANGE
Source file:	CHANGE.FR
Description:	This routine draws the target and trails after the routine PICUP has set up the intensities for them. It also builds the long trail history for use at the end of a run.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	IMAGES, SETIT, FADOFF
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	CHK : check for elevation/azimuth to allow for sweep pattern (if ever added) 0 : both azimuth and elevation 1 : azimuth 2 : elevation
Output arguments:	None
Local variables:	INT : intensity to be used on target and trail TEMPX : temporary for PCX1 to release common for further use TY1 : temporary for PCY1 TY2 : temporary for PCY2 TY3 : temporary for PCY3 TY4 : Temporary for PCY4
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	CREATE
Source file:	CREATE.FR
Description:	This routine calls appropriate routines in the MEGATEK graphics library to create the PAR display and related pictures in the form of a MEGATEK display list.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	IMAGES
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	IFW : dummy argument for a call to a MEGATEK routine which puts a subroutine return in the display list (BRETN) XSCAL : the ratio of screen coordinates to user coordinates in the MEGATEK display in the X - plane. YSCAL : same as above in the Y - plane.
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	FADOFF
Source file:	FADOFF.FR
Description:	This routine is called at the end of a run to fade the trails and turn on the long trail pictures.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	IMAGES
Cancelled by:	N/A
Activates/calls:	CHANGE
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	CHK : argument to CHANGE to determine whether to change azimuth or elevation display or both (if CHK = 0)
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	IMAGES
Source file:	IMAGES.FR
Description:	This routine is the display executive. It turns pictures on and off, initiates the display update and initialization and calls the routines to turn the display processor on and off.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	STARTF, PICUP
Cancelled by:	N/A
Activates/calls:	CREATE, SETIT, FADOFF, SERVUP, CHANGE, WNDCHG, SERVO
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	I : Message code J : Picture number or 0
Output arguments:	None
Local variables:	K : temporary storage
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	OKTOUSEMEGATEK
Source file:	OKTOUSEMEGATEK.FR
Description:	This routine is used to prevent more than one task from accessing the non-reentrant graphics library routines at a time.
Classification:	Logical function
Period:	None
Language:	F
Activated/called by:	STARTF, SERVUP
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	IDUM : dummy function argument
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	Caution! This routine sets the MEGATEK use lock before returning. The user must clear this lock after completion of calls to the graphics library routines.

Title:	PICUP
Source file:	PICUP.FR
Description:	This routine sets up the intensities for the targets and trails sent by radar to be displayed by the display. It also fills the variables in common to be used by the servo routine.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	STARTF
Cancelled by:	N/A
Activates/calls:	IMAGES
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SERVO
Source file:	SERVO.FR
Description:	This routine activates and moves the servo as indicated by joystick position or by explicit request to a position desired by the executives.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	IMAGES, SERVUP, STARTF, SETIT
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SERVUP
Source file:	SERVUP.FR
Description:	This routine retrieves the updated servo position and sends it to CPU 1.
Classification:	Task
Period:	.5 second
Language:	F
Activated/called by:	IMAGES
Cancelled by:	GCA-CTS termination
Activates/calls:	OKTOUSEMEGATEK, TSKERRDLY, SERVO
IPB ID's used:	IDRADAR
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	IPEN : 0: pen up; 1: pen down X,S : joystick position IDUM : dummy function argument
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SETIT
Source file:	SETIT.FR
Description:	This routine resets initial values on MEGATEK pictures. It also reinitializes the servo.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	IMAGES
Cancelled by:	N/A
Activates/calls:	CHANGE, SERVO
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	WNDCHG
Source file:	WNDCHG.FR
Description:	This routine updates the wind display.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	IMAGES
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	I, J, K : temporaries CT : divisor TMPDIR : working wind direction TMPSPD : working wind speed DIR : array of wind direction digits SPD : array of wind speed digits
Files created/changed:	None
Files referenced:	None
Notes:	None

RANGE AND TIME EXECUTIVE

Title:	EXEC
Source file:	EXEC.SR
Description:	This executes any subroutine passed to it as an agrument.
Classification:	Subroutine
Calling sequence:	CALL EXEC (LOCATION), where LOCATION contains the entry address of the routine
Period:	None
Language:	A
Activated/called by:	TIMCAL, RNGCAL, PST1, PSPCH, PWAVE, PTURN, PSPEC
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	ARG0 : address of routine to be executed
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine simulates an .NCAL. This implementation was required to work around a rev 5.21 FORTRAN LOCO bug.

Title:	PCHK
Source file:	PCHK.FR
Description:	This is designed to schedule the placing of SUS phrase numbers in common words and the execution of related subroutines. That is, PCHK is called with the starting range at which a phrase is acceptable, ending range, SUS message, and an index into CTRLR.CO.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	MODELINIT
Cancelled by:	N/A
Activates/calls:	RNGSCHD
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	START : beginning of window during which phrase is acceptable STOP : end of window IDX : index into CTMSG where message goes MESSAGE : to put into CTMSG(IDX) TSK : routine to be called at end of window
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PLACE
Source file:	PLACE.SR
Description:	This puts a subroutine entry address into a word of memory.
Classification:	Subroutine
Calling sequence:	CALL PLACE (SUBROUTINE, LOCATION)
Period:	None
Language:	A
Activated/called by:	TIMSCHD, RNGSCHD
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	SUBROUTINE (ARG0) : the address of the subroutine LOCATION (ARG1) : the place where the subroutine address is to be placed.
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	RNGCAL
Source file:	RNGCAL.FR
Description:	This calls all routines and handles all SKPUT at the top of the range linked-list whose range has been reached.
Classification:	Subroutine
Period:	.5 secs during Phase 2 and 3, after each record during scoring
Language:	F
Activated/called by:	RZEC, PZEC
Cancelled by:	N/A
Activates/calls:	EXEC
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	RANGE : present range
Output arguments:	None
Local variables:	IRANGE : integer range in miles*100
Files created/changed:	None
Files referenced:	None
Notes:	See RNGSCHD for initialization

Title:	RNGSCHD
Source file:	RNGSCHD.FR
Description:	This maintains a linked list of things to be done based on range from touchdown. If the action is to put a SUS phrase number in common then SKPUT is entered as the subroutine entry. Otherwise PLACE is called to put the subroutines entry point there. Note that any routines to be scheduled must be declared EXTERNAL in the routine which calls RNGSCHD.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PCHK, P08, P10A, P13A, PI09, PI10, PI12, PI18, CKRNG, MODELINIT, APGP, PMSCHD
Cancelled by:	N/A
Activates/calls:	PLACE
IPB ID's used:	None
Routines scheduled:	Any passed to it as an argument
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	RANGE : at which SKPUT or routine is to be done ENTRY : SKPUT or the address of routine to be scheduled IDX : index into CTMSG MESSAGE : to put into CTMSG (IDX)
Output arguments:	None
Local variables:	PTR : to move around list LAST : to remember last guy looked at IRANGE : range in miles*100
Files created/changed:	None
Files referenced:	None
Notes:	Initialization : SKRNX=1 SKRNG(1)=-1000 SKNXR(2) thru SKNXR(SKTASKNUM)=-1

Title:	RZEC
Source file:	RZEC.FR
Description:	This executive calls the range and time call executives every cycle, and also PMS in phase 2. In general, it handles the lower priority periodic processing.
Classification:	Task
Period:	.5 second
Language:	F
Activated/called by:	P1AC, P1PRM, P23SUB
Cancelled by:	Self
Activates/calls:	APGP, RDACT, RNGCAL, TIMCAL, STUDTALK, PMS, TGT50, EX1PERT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	BXRZ
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	MSG : intertask message from APE IER : error code
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	TIMCAL
Source file:	TIMCAL.FR
Description:	This activates any subroutines from the top of the time linked-list whose time has come.
Classification:	Subroutine
Period:	.5 sec during Phase 2, after each record during scoring.
Language:	F
Activated/called by:	RZEC, PZEC
Cancelled by:	P3TRM
Activates/calls:	EXEC Any routines in list whose time has come.
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	SAVE : a temporary
Files created/changed:	None
Files referenced:	None
Notes:	See TIMSCHD for initialization

Title:	TIMSCHD
Source file:	TIMSCHD.FR
Description:	This maintains a linked list of routines to be called after an interval of time. It deletes the routine from the queue if the time passed to it is -1. If it is not -1, TIMSCHD looks for a free spot in the queue. If it finds one it inserts the routine where it belongs in the list. If the queue is full it tells this to the bug file and returns. Not that any routine to be scheduled must be declared external in the routine which calls TIMSCHD.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	APE routines : CONCEIVETHE, PLTWAVESHI, MOVEPILOT PMS routines : CKAGP, CK120, OLTCK, PHOSC, PPANEL, PSPEC, P01B, P02A, P02B, P02C, P04C, P05, P05SCH, P06, P11A, P12A, P12C, P14SCH, P15SCH, P17SCH Model controller routines : APGP, BUTX, CLRBUTX, CLREQ, CONTOW, ENDFEED, FEED, FINCON, GIMMIE, HOSAY, IGNORE, IMOFF, LOST, MODELINIT, MSGPICKED, NOACK, OLT, PICKY, SAYIT, TGT50, TOWER, WAVE Other routines : PANEL, RADAR
Cancelled by:	N/A
Activates/calls:	PLACE
IPB ID's used:	None
Routines scheduled:	Any passed to it as argument
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	TIME : when subroutine is to be activated; if -1, delete subroutine ROUTINE : subroutine to be scheduled or deleted
Output arguments:	None
Local variables:	PTR : for looking through list LAST : to remember last guy looked at
Files created/changed:	None
Files referenced:	None
Notes:	Initialization : SKTIME(1)=32000 SKTNX=1 SKNXT(2) thru SKNXT(SKTASKNUM)=-1

MODEL CONTROLLER

Title:	APGP
Source file:	APGP.FR
Description:	This routine calculates the point of interception of the aircraft with the glideslope. Using this information and the speed of the aircraft, the routine determines whether the "approaching glidepath" advisory should be spoken. If so, it puts the advisory into controller common to be spoken by the final controller.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RZEC
Cancelled by:	N/A
Activates/calls:	TIMSCHD, RNGSCHD, ACTOUT
IPB ID's used:	None
Routines scheduled:	BEGDES, NOACK, ENDAPGP
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	X : X distance to glideslope Y : Y distance to glideslope INDEX : index to the array containing the proper distance to the glideslope, dependent on the speed of the plane at which to speak the advisory
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	BEATIT
Source file:	BEATIT.FR
Description:	This routine determines whether or not a waveoff due to lack or cancellation of clearance should occur. If so, it puts the appropriate waveoff message in controller common to be spoken by the final controller and changes the phase of flight for speech recognition.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RNGCAL (by MODELINIT)
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	BEGDES
Source file:	BEGDES.FR
Description:	This routine puts the advisory "begin descent" into controller common.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by APGP)
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	"Begin descent" is removed from controller common when it is no longer legal to speak. Variables are filled that STUDTALK looks at to determine if message is still legal.

Title:	BUTX
Source file:	BUTX.FR
Description:	This routine waits for "Radar button X" to be spoken if the trainee is conducting the approach. If a demonstration is in progress, the routine sends the correct phrase through GLIB to the appropriate speech output device.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by TGT50)
Cancelled by:	N/A
Activates/calls:	GLIB, TIMSCHD
IPB ID's used:	None
Routines scheduled:	GIMMIE, FINCON, HEYFEED, HOLD
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	TEMP : holds VOTRAX phrase to go into SHUSH common block
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	CLEAR
Source file:	CLEAR.FR
Description:	This routine adds the type of clearance the aircraft has received to the advisory to be sent to GLIB for speech output.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	MODWIND
Cancelled by:	N/A
Activates/calls:	MSGPICKED
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	CLRBUTX
Source file:	CLRBUTX.FR
Description:	This routine puts the "button X clear " advisory into controller common.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by CONTOW)
Cancelled by:	ENDFEED
Activates/calls:	TIMSCHD, PANOUT
IPB ID's used:	None
Routines scheduled:	HEYTZEC
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	CLREQ
Source file:	CLREQ.FR
Description:	This routine is the model controller clearance request behavior simulator.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RNGCAL (by MODELINIT)
Cancelled by:	N/A
Activates/calls:	PANOUT, TIMSCHD
IPB ID's used:	None
Routines scheduled:	TOWER
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	CLRNC
Source file:	CLRNC.FR
Description:	This routine simulates the issuance of clearance from the tower by turning on the clearance light on the GCA trainee panel.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by TOWER)
Cancelled by:	N/A
Activates/calls:	PANOUT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	CONTOW
Source file:	CONTOW.FR
Description:	This routine puts the "contact tower after landing" advisory into controller common.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by OLT)
Cancelled by:	N/A
Activates/calls:	TIMSCHD
IPB ID's used:	None
Routines scheduled:	CLRBUTX
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	CSOVER
Source file:	CSOVER.FR
Description:	This routine completes the message created by the final controller model in the routine PICKY by adding the aircraft's call sign and "over" if necessary and by terminating the completed message correctly for speech output.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PICKY
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	DECK
Source file:	DECK.FR
Description:	This routine puts the decision height advisory in controller common. It also determines whether a waveoff is needed and if so, puts the correct waveoff message into controller common. Finally, it changes the flight phase for speech recognition.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RNGCAL (by MODELINIT)
Cancelled by:	N/A
Activates/calls:	ACTOUT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	DESEL
Source file:	DESEL.FR
Description:	This routine is called by FEED to update trainee panel light displays.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	FEED
Cancelled by:	N/A
Activates/calls:	PANOUT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ITMP : delay time, msec K1, K2, K3 : light selections
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	ENDAPGP
Source file:	ENDAPGP.FR
Description:	This routine removes "approaching glidepath" from CTRLR common when the minimum distance at which the advisory is legal has passed.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RNGCAL (by APGP)
Cancelled by:	N/A
Activates/calls:	ACTOUT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	ENDFEED
Source file:	ENDFEED.FR
Description:	This routine is the simulator for the communication between the final controller and the pattern controller in the case of a waveoff or normal termination of a low approach or touch and go.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	IMOFF
Cancelled by:	Self, P2FRZ, RUNKILL
Activates/calls:	TIMSCHD, SAYIT, ACTOUT, GLIB, PANOUT, GO, TSKERRDLY
IPB ID's used:	None
Routines scheduled:	HOLD, HEYTZEC
Cancels:	IMOFF, CLRBUTX
Mailboxes used:	BXFED
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	CT : counter
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	EX1PERT
Source file:	EX1PERT.FR
Description:	This routine enables ISAY to cause EXPERT to be called where the start of a trainee advisory is detected.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	RZEC
Cancelled by:	Self
Activates/calls:	EXPERT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	EXPERT
Source file:	EXPERT.FR
Description:	This routine calls routines to fill controller common with turn and glidepath/course position/trend advisories. It also calls the routine to check recognized trainee input during Phase 3 runs and calls the routine to generate appropriate advisories during demonstrations.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	EX1PERT
Cancelled by:	N/A
Activates/calls:	PICKY, HOWHIGH, HOWFAR, TURN, TRN, NOGYRO
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	FEED
Source file:	FEED.FR
Description:	This routine simulates a pattern controller during the time the aircraft is initially handed off to the final controller. It includes transmitting instructions to the pilot and handoff.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	MODELINIT, P1PRM, P23SUB
Cancelled by:	IMOFF, P2FRZ, RUNKILL
Activates/calls:	PANOUT, POSROG, SAYIT, TIMSCHD, ROGER, GLIB, ACTOUT, DESEL, HOSAY, TSKERRDLY
IPB ID's used:	None
Routines scheduled:	HOLD, HEYTZEC, STPILOT
Cancels:	None
Mailboxes used:	BXFED
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ITMP : temporary for the source of a transmission
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	FINCON
Source file:	FINCON.FR
Description:	This routine puts the initial final controller-pilot contact into controller common.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by BUTX, GIMMIE)
Cancelled by:	N/A
Activates/calls:	GLIB
IPB ID's used:	None
Routines scheduled:	WHEELS
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	GIMMIE
Source file:	GIMMIE.FR
Description:	This routine is scheduled when the pattern controller simulator will not release the frequency. When the trainee is executing an approach, this routine listens for him to ask for the frequency. When a demonstration is in progress, this routine transmits the appropriate message to the speech output device through GLIB.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by BUTX)
Cancelled by:	N/A
Activates/calls:	GLIB, TIMSCHD
IPB ID's used:	None
Routines scheduled:	FINCON
Cancels:	None
Mailboxes used:	BXFED
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	GO
Source file:	GO.FR
Description:	If a waveoff, low approach or touch-and-go is terminating and a demonstration is in progress, this routine is called to give the handoff to the pattern controller. The routine also releases the communication frequency to the pattern controller.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	ENDFEED
Cancelled by:	N/A
Activates/calls:	GLIB, PANOUT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	TRNS : temporary range to fix map position RNG : VOTRAX phrase for map position
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	GTREND
Source file:	GTREND.FR
Description:	This routine determines the proper glidepath trend, if any, dependent on the present and immediately prior aircraft zone, and places its VOTRAX phrase number in controller common.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	HOWHIGH
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	HEYFEED
Source file:	HEYFEED.FR
Description:	This routine sends an intertask message to FEED to inform it that "Radar button X" was spoken.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by BUTX)
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	BXFED
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	HEYTZEC
Source file:	HEYTZEC.FR
Description:	This routine wakes up the training executives at the end of a run.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by CLRBUTX, ENDFEED, FEED, MODELINIT)
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVPHZ
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	HOLD
Source file:	HOLD.FR
Description:	This routine transmits a message to the pattern controllers if the trainee did not give a correct response within the given time limits.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by BUTX, ENDFEED, FEED, HOSAY)
Cancelled by:	IMOFF, SAYIT
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	FEED
Mailboxes used:	BXFED
Events referenced:	EVPHZ
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	HOSAY
Source file:	HOSAY.FR
Description:	Common HOLD scheduling and SAYIT tasking logic.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	FEED
Cancelled by:	N/A
Activates/calls:	TIMSCHD, SAYIT
IPB ID's used:	None
Routines scheduled:	HOLD
Cancels:	None
Mailboxes used:	BXFED
Events referenced:	None
Input arguments:	MSG : SAYIT argument TIME : timeout $: abnormal return on timeout
Output arguments:	None
Local variables:	REPLY : indicator of student input or timeout
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	HOWFAR
Source file:	HOWFAR.FR
Description:	This routine determines the present position of the aircraft on centerline and puts the VOTRAX phrase number for this position into controller common.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	EXPERT
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	HOWHIGH
Source file:	HOWHIGH.FR
Description:	This routine determines the aircraft's present position on glidepath and puts the corresponding VOTRAX phrase number into controller common.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	EXPERT
Cancelled by:	N/A
Activates/calls:	GTREND
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	HOWNOW
Source file:	HOWNOW.FR
Description:	This routine puts the message "How do you hear me now?" into controller common whenever the pilot responds "weak but clear."
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SPEAKPILOT
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	IGNORE
Source file:	IGNORE.FR
Description:	This routine changes the trainee panel to ignore a clearance request issued by the final controller.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by TOWER)
Cancelled by:	N/A
Activates/calls:	PANOUT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	LOST
Source file:	LOST.FR
Description:	This routine puts radar contact lost in CTRLR common.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by RADAR)
Cancelled by:	N/A
Activates/calls:	ACTOUT, TIMSCHD
IPB ID's used:	None
Routines scheduled:	IMOFF
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	LOW
Source file:	LOW.FR
Description:	This routine puts the low altitude alert message in controller common.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by MOVEPILOT)
Cancelled by:	N/A
Activates/calls:	ACTOUT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	MODELINIT
Source file:	MODELINIT.FR
Description:	This routine performs the initialization for the controller models.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PB23SUB, P1PRM
Cancelled by:	N/A
Activates/calls:	PCHK, RNGSCHD, PMSCHD, VARIMOD, PANOUT, SELBUT, TIMSCHD
IPB ID's used:	None
Routines scheduled:	DECK, OLT, BEATIT, PULLRANGE, HEYTZEC, CLREQ
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ITMP : temporary for the source of speech output
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	MODWIND
Source file:	MODWIND.FR
Description:	This routine adds the wind advisory to the clearance advisory built by PICKY, the model final controller.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PICKY
Cancelled by:	N/A
Activates/calls:	CLEAR
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	MSGFILL
Source file:	MSGFILL.FR
Description:	This routine is scheduled to inform APE to look in the SHUSH buffer for a model controller transmission. This prevents the pilot from acting upon a transmission before the synthesizer is finished speaking it.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by MSGPICKED)
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	I : locp index
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	MSGPICKED
Source file:	MSGPICKED.FR
Description:	This routine decides which SHUSH buffer to use for the current model controller transmission.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	CLEAR, PICKY, STOPTURN
Cancelled by:	N/A
Activates/calls:	TIMSCHD
IPB ID's used:	None
Routines scheduled:	MSGFILL
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	CTER : number of attempts to fill the SHUSH buffer
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	NOACK
Source file:	NOACK.FR
Description:	This routine puts the "do not acknowledge further transmissions" into controller common.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by APGP)
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	NOGYRO
Source file:	NOGYRO.FR
Description:	This routine determines whether a no-gyro situation exists and if so, fills common with the VOTRAX phrase number to inform the pilot.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	EXPERT
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	OLT
Source file:	OLT.FR
Description:	This routine puts the "over landing threshold" advisory into controller common, determines the aircraft's position on centerline and puts this additional information in common.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RNGCAL (by MODELINIT)
Cancelled by:	N/A
Activates/calls:	ACTOUT
IPB ID's used:	None
Routines scheduled:	CONTOW, IMOFF
Cancels:	IMOFF
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	MILES : range in integer miles * 100
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PICKY
Source file:	PICKY.FR
Description:	Working from the list of acceptable messages in controller common, this routine chooses the highest priority message, completes it, and sends it to GLIB for speech output.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	EXPERT
Cancelled by:	N/A
Activates/calls:	GLIB, MODWIND, TIMSCHD, CSOVER, POSOLT, MSGPICKED, POSADH
IPB ID's used:	None
Routines scheduled:	STOPTURN
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ALARMCHECK : T if waveoff alarm is on
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	POSADH
Source file:	POSADH.FR
Description:	This routine chooses the appropriate model controller message(s) at decision height.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PICKY
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	$: return taken unless a waveoff is to be given
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	POSOLT
Source file:	POSOLT.FR
Description:	This routine determines course position at landing threshold.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PICKY
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	POSROG
Source file:	POSROG.FR
Description:	This routine acknowledges the pattern controller handoff and turns on the appropriate buttons on the trainee panel.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	FEED
Cancelled by:	N/A
Activates/calls:	GLIB, SELBUT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ITMP : temporary for source of speech output
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PULLRANGE
Source file:	PULLRANGE.FR
Description:	This routine removes the milemark advisories from controller common when the maximum legal bound for transmission of this advisory has been exceeded.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RNGCAL (by MODELINIT)
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PUTWIND
Source file:	PUTWIND.FR
Description:	This routine puts the wind advisory in controller common.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TOWER
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	ROGER
Source file:	ROGER.FR
Description:	This routine responds to the pattern controller when the controller is giving instructions.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	FEED
Cancelled by:	N/A
Activates/calls:	GLIB
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ITMP : temporary for source of speech output
Files created/changed:	None
Files referenced:	None
Notes:	None

AD-A087 190 LOGICON INC SAN DIEGO CA TACTICAL AND TRAINING SYSTE--ETC F/G 17/9
GROUND CONTROLLED APPROACH CONTROLLER TRAINING SYSTEM (GCA-CTS)--ETC(U)
JUN 80 G BARBER, J BOLLENBACHER, D BREWTON N61339-77-C-0162

UNCLASSIFIED NAVTRAEQUIPC-77-C-0162-3 NL

5 OF 8

AD
A087190

Title:	SAYIT
Source file:	SAYIT.FR
Description:	This routine waits for voice input from trainee and transmits a message if the input is the one expected by the pattern controller.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	FEED, ENDFEED
Cancelled by:	HOLD, Self, IMOFF, RUNKILL
Activates/calls:	TIMSCHD
IPB ID's used:	None
Routines scheduled:	None
Cancels:	HOLD
Mailboxes used:	BXFED
Events referenced:	EVVIN
Input arguments:	None
Output arguments:	None
Local variables:	ITMP : temporary for trainee's speech recognized phrase
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SELBUT
Source file:	SELBUT.FR
Description:	This routine accesses the panel driver for the model controller.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	FEED, MODELINIT, POSROG
Cancelled by:	N/A
Activates/calls:	PANOUT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	TYPE : which button to turn on
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	STOPTURN
Source file:	STOPTURN.FR
Description:	This routine transmits the "stop turn" advisory when a nogyro turn is in progress.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by PICKY)
Cancelled by:	N/A
Activates/calls:	GLIB, MSGPICKED
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	STPILOT
Source file:	STPILOT.FR
Description:	Starts APE after the handoff is complete.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by FEED)
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	STUDTALK
Source file:	STUDTALK.FR
Description:	This routine takes speech recognized advisories from the trainee and removes them from the controller list of acceptable messages. It also checks for messages on the queue which are no longer legal and removes them from the queue.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RZEC
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	PHRASE : phrase spoken by the trainee CHECK : elapsed time used to see if a phrase should should be removed from the queue
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	TGT50
Source file:	TGT50.FR
Description:	This routine looks for 50% of azimuth target to appear on the radar screen.
Classification:	Subroutine
Period:	None
Language:	F
Activated called by:	RZEC
Cancelled by:	None
Activates/calls:	ACTOUT, TIMSCHD
IPB ID's used:	None
Routines scheduled:	BUTX
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	TEMP : temporary variable
Files created/changed:	None
Files referenced:	None
Notes:	This routine need be scheduled only if handoff is to made.

Title:	TOWER
Source file:	TOWER.FR
Description:	This routine is the tower controller simulator. It processes clearance requests from the final controller. It also causes the wind simulation to freeze with wind conditions that are in the list of recognizable phrases.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by PANEL, CLREQ)
Cancelled by:	N/A
Activates/calls:	TIMSCHD, ACTOUT, PUTWIND
IPB ID's used:	None
Routines scheduled:	WAVE, CLRNC, IGNORE
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	RANGE : range in miles OUTIME : time to send msg for clear WVTM : time to send msg for waveoff
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	TRN
Source file:	TRN.FR
Description:	This model controller routine vectors the aircraft on final.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	EXPERT
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	TURN
Source file:	TURN.FR
Description:	This is the model controller turn-advisory generator for the turn to final.
Classification:	Subroutine
Period:	None
Language:	F
Activated called by:	EXPERT
Cancelled by:	None
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referanced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ISIGNX : 1.0 times sign of current ACX BLIPH : real-space height of current target blip (feet) OL : target centerline overlap as pct of BLIPH ABSOL : absolute value (OL) CZONE : current A/C "turn zone" TMOD : current CTHEAD, mod 5 NHEAD : new ideal heading TMPACHDEG : ACH in degrees
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	VARIMOD
Source file:	VARIMOD.FR
Description:	This routine initializes common variables for the model controller.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	MODELINIT, P1AC
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	WALOFF
Source file:	WALOFF.FR
Description:	This routine turns out the waveoff light for the model controller.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by WAVE)
Cancelled by:	N/A
Activates/calls:	PANOUT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	WAVE
Source file:	WAVE.FR
Description:	This routine changes the trainee panel to indicate waveoff conditions.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by TOWER)
Cancelled by:	N/A
Activates/calls:	PANOUT, TIMSCHD
IPB ID's used:	None
Routines scheduled:	WALOFF
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	WHEELS
Source file:	WHEELS.FR
Description:	This routine puts the "wheels down" advisory in controller common, if the pilot hasn't already done so.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by FINCON)
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

DIGITIZED SPEECH

Title:	SBF
Source file:	SBF.SR
Description:	This provides an entry only. It marks the logical address of the start of digitized speech buffers.
Classification:	Entry point
Period:	None
Language:	A
Activated/called by:	N/A
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SPBUF
Source file:	SPBUF.FR
Description:	This reads recorded digitized speech from the disk into core buffers for device 31.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	SYSINIT
Cancelled by:	DIE
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	BXPLY, BXSPH
Events referenced:	EVSPT
Input arguments:	None
Output arguments:	None
Local variables:	GUNK : to pass to BXPLY IER : error argument
Files created/changed:	None
Files referenced:	RPLSPH, RPPSPH, CANFILE, IDVFILE
Notes:	None

Title:	SPDMP
Source file:	SPDMP.FR
Description:	This dumps the core buffers filled by device 31 into a predefined location on the disk. This location is determined by SPIN.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	SYSINIT
Cancelled by:	DIE
Activates/calls:	ACTOUT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	BXPLY, BXSPH, BXRC
Events referenced:	EVSPN
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	RPLSPH, RPPSPH, CANFILE, DVFILE
Files referenced:	None
Notes:	None

Title:	SPFR
Source file:	SPDR.SR
Description:	This routine stops the speech digitizer without changing its mode.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	ERRHAN, KTEACH
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SPGO
Source file:	SPDR.SR
Description:	This routine restarts digitized voice without changing its mode.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	ERRHAN, KTEACH
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SPIN
Source file:	SPIN.FR
Description:	SPIN determines where on the disk new digitized speech recording should go and calls STRTREC to start the device.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	DIGIN, P23SUB
Cancelled by:	None
Activates/calls:	STRTREC
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	NUM : number of the phrase to be recorded RECPOS: place recording should go = 0 : at next available slot = 1 : over last recording
Output arguments:	None
Local variables:	IER : error argument
Files created/changed:	None
Files referenced:	IDVFILE, PDVFILE
Notes:	None

Title:	SPNIT
Source file:	SPDR.SR
Description:	This routine informs RDOS about the speech digitizer and where its interrupt service routine is.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	SYSINIT
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SPOFF
Source file:	SPDR.SR
Description:	This routine turns the speech digitizer off by removing device 31 from the RDOS interrupt structure.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	DIE
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	SPCD : device code for digitizer
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SPOUT
Source file:	SPOUT.FR
Description:	This prepares the speech digitizer for output of recorded speech. It accepts an argument which tells it if this is the replay mode. If it is not the relay mode, it determines if this is 'canned' and responds accordingly and starts the device playing. If this is replay, it sets up the playback to start at the beginning of the speech file but does not start the device.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	DIGIN, REPLAY, DONE, P1PRM
Cancelled by:	None
Activates/calls:	STRTPLY
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	BXPLY
Events referenced:	None
Input arguments:	NUM : Ø if replay else phrase number to output
Output arguments:	None
Local variables:	GUNK : to pass to BXPLY
Files created/changed:	None
Files referenced:	IDVFILE, PIDVFILE
Notes:	None

Title:	STRTPLY
Source file:	SPDR.SR
Description:	This starts digitized voice playing by turning on device 31 in the play mode.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	REPLAY, SPOUT
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	STRTREC
Source file:	SPDR.SR
Description:	This starts digitized recording by turning on device 31 in the record mode.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	SPIN
Cancelled by:	STRTPLY
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

SYNTHESIZED SPEECH

Title:	DONE
Source file:	DONE.FR
Description:	This task handles GLBF's VOTRAX or audio output. If a message is in the output queue, it is sent to the appropriate device controller.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	GLBF
Cancelled by:	Self
Activates/calls:	SPOUT, ACTOUT, VSOUT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVSPT, EVVRO
Input arguments:	None
Output arguments:	None
Local variables:	I,J : loop indices IBLK : ACTOUT 8 word blk. counter IPTR : temp. queue ptr. ITEMP : array(16). VSCON/ACTOUT call argument array. ITME : output time IER : error code
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	GLBF
Source file:	GLBF.FR
Description:	This routine qualifies and schedules all message output from the final controller, pattern controller and pilot models. It also handles prompt requests during voice data collection and validation and Phase 1. The messages are queued for output to avoid message overlaps.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	GLIB
Cancelled by:	N/A
Activates/calls:	DONE
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	IFRM : message source 0 : bypass source check 1 : final controller 2 : feeder controller 3 : pilot IDEV : message destination device IDIM : size of message array IARRAY : array start
Output arguments:	IFRM : set to -1 if message cannot be transmitted
Local variables:	IER : ERROR I : loop index
Files created/changed:	None
Files referenced:	None
Notes:	GLBF should be initially compiled with /X option to note any need for a longer queue. The note is written via NCBUG.

Title:	GLIB
Source file:	GLIB.SR
Description:	This routine examines the call arguments and restructures them to conform to GLBF (the FORTRAN 5 version of GLIB) call parameters.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	FEED, PICKY, VSPRES, ACTIVITY, ENDFEED, GO, BUTX, GIMMIE, FINCON, POSROG, SRMON, ROGER, SPEAKPILOT, STOPTURN
Cancelled by:	None
Activates/calls:	GLBF
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	Address pointers to: SOURCE : message source DESTIN : destination output device TYPE : argument list type 0 array or -1 list LIST : array start or end of list
Output arguments:	None
Local variables:	ARRAY : array start address COUNT : argument counter DEMS : array dimension ARRAY : stack disp. COUNT : loop control K5 : argument count : subtraction constant K1 : decrement/increment by 1
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	RDFRAZ
Source file:	RDFRAZ.SR
Description:	This task is activated to read the phoneme file for VOTRAX into buffers that it shares with WRFRAZ. It attempts to get ahead of WRFRAZ as much as possible.
Classification:	Task
Period:	None
Language:	A
Activated/called by:	WRFRAZ
IPB ID's used:	None
Routines scheduled:	None
Cancelled by:	Self
Activates/calls:	None
Cancels:	Self
Mailboxes used:	Gets one as argument from WRFRAZ
Events referenced:	None
Input arguments:	ITOTL : the number of phrases desired IFRAZ : the buffer with phrase numbers to be read
Output arguments:	IRED : number of phrases read .MSG : mailbox to tell WRFRAZ it's done
Local variables:	IWRTMP : buffer being written to
Files created/changed:	None
Files referenced:	FRAZ.VO
Notes:	None

Title:	VSOUT
Source file:	VSOUT.FR
Description:	This routine takes a buffer filled by DONE and decodes the contents, replacing them by the VOTRAX phrase numbers which are appropriate.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	DONE
Cancelled by:	N/A
Activates/calls:	WRFRAZ
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	IPHRZ : a buffer of arguments converted to phrase numbers and output IFIL : the size of IPHRZ
Output arguments:	None
Local variables:	NEXT : pointer into IPHRZ IQX : keeps track of number of phrases put into IQPHX IQPHX : buffer of phrase numbers to be sent to RDFRAZ I : Loop control IER : Error argument
Files created/changed:	None
Files referenced:	None
Notes:	The priority of VSOUT must be above 15 and VRO must be open to channel 5

Title:	WRFRAZ
Source file:	WRFRAZ.SR
Description:	This routine accepts a buffer full of phrase numbers from VSOUT, tasks RDFRAZ to fill another buffer with the corresponding phonemes from FRAZ.VO, and then writes that buffer to the VOTRAX device.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	VSOUT
Cancelled by:	N/A
Activates/calls:	RDFRAZ
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	Sends one as an argument to RDFRAZ
Events referenced:	None
Input arguments:	ITOTL : number of phrases to be output IFRAZN : buffer which contains phrases;
Output arguments:	None
Local variables:	IRED : number of phrases read by RDFRAZ IWRIT : number of phrases written MSG : mailbox to pass to RDFRAZ BUF1, BUF2, BUF3 : buffers for RDFRAZ to write into
Files created/changed:	None
Files referenced:	$VRO
Notes:	This is a root code subroutine to VSOUT. It provides multi buffering capability of FRAZ file reading and $VRO writing.

PERFORMANCE MEASUREMENT AND SCORING

Title:	AFAPGP
Source file:	AFAPGP.FR
Description:	This routine checks for an "over" after approaching glidepath and for its correctness.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS if PVNEX(3) set
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine is scheduled by P04A. It sets bits 1, 2 of PV04.

Title:	AFDNA
Source file:	AFDNA.FR
Description:	This routine checks for the presence of "over" after "do not acknowledge further advisories".
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS via PVNEX(2) set by P04B
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bit 7 of PV04.

Title:	AFWC
Source file:	AFWC.FR
Description:	This routine records error if phrase after wheel check is not "over."
Classification:	Subroutine
Period:	None
Language:	F
Activated called by:	P04D through PSUS
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bit 15 of PV04.

Title:	CK120
Source file:	CK120.FR
Description:	This routine detects failure to give counter corrective turn within 8 seconds of a turn > 120.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by P05)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	P05
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bit 4 of PV05.

Title:	CKACK
Source file:	CKACK.FR
Description:	This routine checks for omission of handoff acknowledgement.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by PSPEC)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine is scheduled by PSPEC for 10 secs. after handoff given. It sets bit 4 of PV01.

Title:	CKADH
Source file:	CKADH.FR
Description:	This is the omission check for "at decision height."
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RNGCAL (by PI09)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bit 8 of PV09.

Title:	CKAGP
Source file:	CKAGP.FR
Description:	This is the omission check for "approaching glidepath".
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSPEC
Cancelled by:	N/A
Activates/calls:	TIMSCHD, PERRCHK
IPB ID's used:	None
Routines scheduled:	CKBD
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERROR : word for error reporting
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bits 3, 13 of PV04 and schedules CKBD for 30 secs. hence, it is not a true CK** routine since it is purely PSPEC processed and not timed.

Title:	CKBD
Source file:	CKBD.FR
Description:	This is the omission check for "begin descent".
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by CKAGP, P04C)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERROR : word for error reporting
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bits 8, 10 of PV04. It clears bit 3 of PV00. It is scheduled by CKAGP 30 secs. after allowable AGP advisory range.

Title:	CKCHK
Source file:	CKCHK.FR
Description:	This routine checks that the radio-check was given within 30 seconds of 50% target appearance.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by PSPEC)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This is scheduled by PSPEC when 50% target record is received. It sets bit 1 of PV02.

Title:	CKCLR
Source file:	CKCLR.FR
Description:	This routine checks for omission of 2nd request to tower.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RNGCAL (by P10A)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine is scheduled by P10A when 1st request is denied, at 1.9 miles from touchdown. It sets bit 3 of PV10.

Title:	CKCN
Source file:	CKCN.FR
Description:	This routine checks that a radar contact was reported within 10 seconds of 50% target appearance.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by PSPEC)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine is scheduled by PSPEC 10 secs. after 50% target. It sets bit 7 of PV01.

Title:	CKCOR
Source file:	CKCOR.FR
Description:	This routine detects failure to give trend or turn within 3 seconds of a "well" azimuth message.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by P05, P06)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERRWD : word for error reporting
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bit 2, 3 of PV06 and uses PV06(2).

Title:	CKCRP
Source file:	CKCRP.FR
Description:	This routine checks for target transiting between azimuth zones 2 and 3.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	CKIN
Cancelled by:	N/A
Activates/calls:	ACTOUT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine's purpose is to write the PMS activity record for this event.

Title:	CKCWO
Source file:	CKCWO.FR
Description:	This routine checks that a waveoff was given under clearance problems.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RNGCAL (by P10A), TIMCAL (by PPANEL)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERRWD : error word for PERRCHK
Files created/changed:	None
Files referenced:	None
Notes:	This routine is scheduled by P10A when 1st clearance denied, at 1.3 miles. Also, PPANEL schedules it at waveoff light on + 2 seconds. It sets bits 10, 12 of PV10.

Title:	CKEZN
Source file:	CKEZN.FR
Description:	This is the omission check of glidepath position calls.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSPEC
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This is called whenever PSPEC encounters a GP position transition record. It sets bit 3 of PV07 and uses PV07 (3,10).

Title:	CKFCP
Source file:	CKFCP.FR
Description:	This routine checks for the omission of the final course position of "over" after the OLT advisory.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by P11A)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bit 3 of PV11. It is scheduled by P11A 3 secs. after OLT said.

Title:	CKGMR
Source file:	CKGMR.FR
Description:	If pattern did not release freq., this routine checks that a "give me..." request was made within 15 seconds.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by P01B)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine is scheduled by P01B through TIMCAL and sets PV01, bit 10.

Title:	CKGPP
Source file:	CKGPP.FR
Description:	This routine checks for target leaving an elevation zone.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	CKIN
Cancelled by:	N/A
Activates/calls:	ACTOUT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine writes PMS records for GP zone transitions.

Title:	CKHDCOR
Source file:	CKHDCOR.FR
Description:	This routine checks that a no-gyro heading correction was given within 20 seconds of a zone 2-3 transition.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by P14SCH)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine is scheduled by P14SCH. It sets bit 3 of PV14. It uses PV14(3, 4, 5) as counters.

Title:	CKHN
Source file:	CKHN.FR
Description:	This routine checks that a "how...now?" is given within 15 seconds of a below-normal radio-check.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by P02C)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine is scheduled by P02A when a below normal level radio-check is given and sets bit 8 of PV02.

Title:	CKHO
Source file:	CKHO.FR
Description:	This routine checks that a handoff was given to the pattern controller within 30 sec. of waveoff, ADH for low appr., and OLT for touch and go appr.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by PHOSCH)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bit 6 of PV12. It is scheduled by PHOSCH.

Title:	CKICS
Source file:	CKICS.FR
Description:	This routine checks for ICS deselected 10 seconds after pattern controller releases frequency.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by PPANEL)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bit 12 of PV01.

Title:	CKIN
Source file:	CKIN.FR
Description:	This routine checks for target being in zones 1 or 2 for 1/2 mile, or 5 miles (2 for short approach), whichever comes first and calls CKCRP when condition has been detected. It also checks for glidepath position advisory propriety and calls CKGPP.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RZEC
Cancelled by:	N/A
Activates/calls:	CKCRP, CKGPP
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine's purpose is to activate CKCRP under proper conditions. CKGPP and CKCRP provide activity records for PMS.

Title:	CKK3
Source file:	CKK3.FR
Description:	This routine checks that the mike was unkeyed within 3 seconds of radio check and "how...now?".
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by P02A, P02B)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine is scheduled by P02A and P02B. It sets bit 6 of PV02.

Title:	CKK5
Source file:	CKK5.FR
Description:	This routine checks that the mike stayed unkeyed at least 5 sec. after being unkeyed in the radio-check procedure.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by PPANEL)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine is scheduled by PPANEL when mike unkeyed and PVMIKE is true. It sets bit 6 of PV02.

Title:	CKLAA
Source file:	CKLAA.FR
Description:	This routine makes the omission check for low altitude alert.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by PSPEC)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bits 1, 2 of PV16.

Title:	CKNGA
Source file:	CKNGA.FR
Description:	This routine checks that a no-gyro approach advisory has been given if needed.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RNGCAL (by P13A)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bit 2 of PV13. It is scheduled by P13A within 3/4 mi of "hdg xxx" warning.

Title:	CKOLT
Source file:	CKOLT.FR
Description:	This routine checks for an "OLT" omission and for "OLT" timing.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by PSPEC)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bit 2 of PV11. It is scheduled by PSPEC 1 second after OLT point.

Title:	CKOVR
Source file:	CKOVR.FR
Description:	This routine checks for omission of "over" after final course position of OLT advisory.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by OLTCK)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine is scheduled by OLTCK, 3 secs. after final position call and sets bit 5 of PV11.

Title:	CKP18
Source file:	CKP18.FR
Description:	This routine checks for transmission rate errors at decision height point.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RNGCAL (by PI18)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERRWD : word for error reporting
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bits 1, 2 of PV18(0) and uses PV18(1, 2, 3) and PV05(9, 10).

Title:	CKPAT
Source file:	CKPAT.FR
Description:	This routine checks that the pattern controller was notified within 10 secs. after rollout instructions were given.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by P12A)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERRWD : word for error reporting
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bits 3, 4 of PV12. It is scheduled by P12A, 10 seconds after rollout given.

Title:	CKPCLR
Source file:	CKPCLR.FR
Description:	This routine checks for the omission of the clearance message to the pilot.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RNGCAL (by PI10)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine is scheduled by initialization at 1 mile. It sets bit 9 of PV10.

Title:	CKRFR
Source file:	CKRFR.FR
Description:	This routine checks that the radio frequency has been released 5 seconds after "C/S radar contact" was received.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by P12C)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine clears bit 10 in PV00 and sets bit 13 in PV12.

Title:	CKRNG
Source file:	CKRNG.FR
Description:	This routine checks for the omission of range calls within the correct range limits.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RNGCAL (by CKRNG)
Cancelled by:	N/A
Activates/calls:	PERRCHK, RNGSCHD
IPB ID's used:	None
Routines scheduled:	CKRNG, CKROM
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	RANGE : aircraft range
Files created/changed:	None
Files referenced:	None
Notes:	This routine is scheduled by P08, CKRNG, and by initialization routine at 4.9 miles. It sets PV08 bits 1-8, 11 and uses PV08(1, 6, 5).

Title:	CKROM
Source file:	CKROM.FR
Description:	This routine checks for the complete omission of range calls.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RNGCAL (by CKRNG)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	RANGE : aircraft range
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets PV08(0) bit 11 and increments PV08.

Title:	CKTB
Source file:	CKTB.FR
Description:	This routine checks for the omission of a transmission break after the "do not ack..." and prior to 1 mile.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RNGCAL (by PI17)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine is scheduled by initialization at 1 mile.

Title: CKTLS
Source file: CKTLS.FR
Description: This routine checks the quality of the turn to final.
Classification: Subroutine
Period: None
Language: F
Activated/called by: PSPEC
Cancelled by: N/A
Activates/calls: PERRCHK
IPB ID's used: None
Routines scheduled: None
Cancels: None
Mailboxes used: None
Events referenced: None
Input arguments: None
Output arguments: None
Local variables: None
Files created/changed: None
Files referenced: None
Notes: If this routine is processed at the 6 mi. (3 mi. for short apr.) record it sets bit 1 of PV03. If it is initiated by the 5 mi. (2 for short apr.) special record it sets bit 2 of PV03.

Title: CKWO
Source file: CKWO.FR
Description: This routine checks for omission of waveoff due to radar contact lost and minimum separation.
Classification: Subroutine
Period: None
Language: F
Activated/called by: TIMCAL (by P15SCH
Cancelled by: N/A
Activates/calls: PERRCHK
IPB ID's used: None
Routines scheduled: None
Cancels: None
Mailboxes used: None
Events referenced: None
Input arguments: None
Output arguments: None
Local variables: None
Files created/changed: None
Files referenced: None
Notes: This routine sets bits 2, 5 of PV15. It is scheduled by P15SCH when special records for conditions are received.

Title:	CKZN3
Source file:	CKZN3.FR
Description:	This routine checks omission of correction with 30 seconds of target entering zone 3.
Classification:	Subroutine
Period:	None
Language:	A
Activated called by:	TIMCAL (by P05SCH, P05)
Cancelled by:	NoA
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bit 5 of PV05 and uses PV05(5).

Title:	DHCK
Source file:	DHCK.FR
Description:	This routine scores position message for ADH when SUS record following ADH received.
Classification:	Subroutine
Period:	None
Language:	F
Activated called by:	PSUS
Cancelled by:	None
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERRWD : word for error reporting
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bit 2 of PV09, bit 10 of PV15, resets bits 3,4,5,6 of PV09 and schedules WOCK via PVNEX(8).

Title:	DIRT
Source file:	DIRT.FR
Description:	This routine returns true if turn given is in correct direction, otherwise returns false.
Classification:	Logical function
Period:	None
Language:	F
Activated/called by:	P03, P14A
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	PHRASE : phrase number of turn PRESHD : present magnetic aircraft hdg. FINHD : final aircraft hdg.
Output arguments:	DIRT - T if turn in correct direction
Local variables:	DIFF - difference between runway and turn hdg.
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	FB19
Source file:	FB19.FR
Description:	This routine provides feedback about the alignment checking procedure.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SC19
Cancelled by:	N/A
Activates/calls:	ERINDEX, EXPLAIN
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	I1 : first bit in PV19 to be checked I2 : last bit in PV19 to be checked
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	HOCK
Source file:	HOCK.FR
Description:	This routine scores handoff procedure when SUS record for "missed approach" is received or phrase following "on the go" is received.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERRWD : word for error reporting
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bits 7, 9, 10, 11, 14 of PV12 and clears PV00 bit 9.

Title:	MARKIT
Source file:	MARKIT.FR
Description:	This routine generates special activity records for critical mile markers for phase 3 runs.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RNGCAL (by PMSCHD)
Cancelled by:	N/A
Activates/calls:	ACTOUT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	MILER
Source file:	MILER.FR
Description:	This routine generates special activity records for 6 and 5 miles (3,2 for short approaches).
Classification:	Subroutine
Period:	None
Language:	F
Activated called by:	RNGCAL (by PMSCHD)
Cancelled by:	N/A
Activates/calls:	ACTOUT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	OLTCK
Source file:	OLTCK.FR
Description:	This routine scores course position and "over" of PV11 when SUS record after "OLT" encountered.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	N/A
Activates/calls:	TIMSCHD, PERRCHK
IPB ID's used:	None
Routines scheduled:	CKOVR
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERRWD : word for error reporting
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bits 4, 5 of PV11. It schedules CKOVR in 3 seconds to check for omission of "over" after final course position given. It also schedules itself via PVNEX(5).

Title:	P01A
Source file:	P01A.FR
Description:	This routine scores the handoff acknowledgement upon receiving SUS record for handoff acknowledgement.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PST1
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine clears PV00, bit 0 and sets PV01, bit 2.

Title:	P01B
Source file:	P01B.FR
Description:	This routine scores the radar contact report when the SUS record for radar contact is received.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PST1
Cancelled by:	N/A
Activates/calls:	TIMSCHD, PERRCHK
IPB ID's used:	None
Routines scheduled:	CKGMR
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine scores bits 3, 6, 8, 9 of PV01.

Title:	P01C
Source file:	P01C.FR
Description:	This routine scores the "give me ..." request SUS record.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PST1
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets PV01, bit 11.

Title:	P01D
Source file:	P01D.FR
Description:	This routine checks for omission of radar contact report when the radio-check is given.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PST1
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bit 5 of PV01.

Title:	P02A
Source file:	PQ2A.FR
Description:	This routine scores the radio check procedure when SUS record for radio-check is received.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PST1
Cancelled by:	N/A
Activates/calls:	TIMSCHD, PERRCHK
IPB ID's used:	None
Routines scheduled:	CKK3
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERRWD : Word for error reporting
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bits 2, 3, 4, 5 of PV02.

Title:	P02B
Source file:	P02B.FR
Description:	This routine scores the speech quality of the radio-check when SUS record for "how...now?" is received.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	N/A
Activates/calls:	TIMSCHD, PERRCHK
IPB ID's used:	None
Routines scheduled:	CKK3
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERRWD : word for error reporting
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bits 3, 8 of PV02.

Title:	P02C
Source file:	P02C.FR
Description:	This routine grades student voice level for radio check.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSPCH
Cancelled by:	N/A
Activates/calls:	TIMSCHD, PERRCHK
IPB ID's used:	None
Routines scheduled:	CKHN
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bits 7, 9 of PV02.

Title:	P03
Source file:	P03.FR
Description:	This routine scores turn to final advisories when the turn to final SUS record is received.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PTURN
Cancelled by:	N/A
Activates/calls:	PERRCHK, DIRT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERRWD : word for error reporting NOTLAST : logical variable : T if heading does not exceed model's last turn to final heading TRNDIF : difference between model and trainee turn heading
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bit 7, 10, 13, 9, 12, 15, 3, of PV03.

Title:	P04A
Source file:	P04A.FR
Description:	This routine scores the "approaching glidepath" advisory for proper time and order.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	ERRWD : word whose bits indicate errors detected by this call.
Local variables:	CALLSIGN : mnemonic for A/C callsign
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bits 2, 3, 4, 13 of PV04 and clears bit 1 of PV00.

Title:	P04B
Source file:	P04B.FR
Description:	This routine scores the "do not acknowledge" advisory for proper time and order.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	PRESSCORE : word whose bits indicate errors detected in this call
Local variables:	CALLSIGN : mnemonic for A/C cal'sign
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bits 5, 6, 8 of PV04 and clears bit 2 of PV00. The routine sets bit 2 in PVNEX to ensure that a routine is initiated to see that "over" is not given.

Title:	P04C
Source file:	P04C.FR
Description:	This routine scores the begin descent advisory for proper time and order.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	N/A
Activates/calls:	PERRCHK, TIMSCHD
IPB ID's used:	None
Routines scheduled:	CKBD
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	ERRWD : word whose bits indicate errors detected by this call
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bits 10, 11, 12 of PV04 and clears bit 3 of PV00.

Title:	P04D
Source file:	P04D.FR
Description:	This routine scores wheel check advisory for proper time and order.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	ERRWD : word whose bits indicate errors detected by this call
Local variables:	CALLSIGN : holds A/C callsign ERRWD : word for error reporting
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bits 14, 15 of PV04 and schedules AFWC via PVNAX(4)

Title:	P05
Source file:	P05.FR
Description:	This routine scores heading advisories.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PTURN
Cancelled by:	N/A
Activates/calls:	TIMSCHD, PERRCHK
IPB ID's used:	None
Routines scheduled:	CKCOR, CK120, CKZN3
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERRWD : word for error reporting
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bits 1, 2, 3, 7, 8 of PV05 and uses PV05(1, 2, 3, 7, 8, 9, 10, 11, 12, 14).

Title:	P05SCH
Source file:	P05SCH.FR
Description:	This routine schedules check for correction 30 secs after the target enters zone 3.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSPEC
Cancelled by:	N/A
Activates/calls:	TIMSCHD
IPB ID's used:	None
Routines scheduled:	CKZN3
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine uses PV05(13).

Title:	P06
Source file:	P06.FR
Description:	This routine scores azimuth position and trend messages.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	N/A
Activates/calls:	TIMSCHD, PERRCHK
IPB ID's used:	None
Routines scheduled:	CKCOR
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERRWD : word for error reporting whose bits indicate errors detected by this call MAZONE : message azimuth zone
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bits 1, 4 of PV06 and uses PV06(1, 5, 4, 6, 7).

Title:	P07A
Source file:	P07A.FR
Description:	This routine checks that "begin descent" was given prior to any glidepath call.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	P07B, P07C
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine is called when GP position or trend is given. It sets bit 1 of PV07 and uses PV07(8), PV07(1).

Title:	P07B
Source file:	P07B.FR
Description:	This routine scores glidepath position calls when SUS record for glidepath position is received.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	N/A
Activates/calls:	PERRCHK, P07A
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERRWD : word for error reporting MAZONE : message glidepath zone
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets PV07 bits 2, 5, 7.

Title:	P07C
Source file:	P07C.FR
Description:	This routine scores glidepath trend calls when SUS record for trend call received.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	N/A
Activates/calls:	PERRCHK, P07A
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERRWD : word for error reporting TND : glidepath trend
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bits 4, 6 of PV07. It also uses PV07(4,6,9,11).

	TREND-CLASS	
MESSAGE	CLIMBING	DESCENDING
Well above glidepath	2	5
Above glidepath	2	5
Slightly above glidepath	0	5
On glidepath	1	5
Slightly below glidepath	1	3
Below glidepath	1	4
Well below glidepath	1	4

Cortrend: Correct trend-message for trend-class:

TREND-CLASS	CORRECT MESSAGE
0	Going above glidepath
1	Coming up
2	Going further above glidepath
3	Going below glidepath
4	Going further below glidepath
5	Coming down

Title:	P08
Source file:	P08.FR
Description:	This routine scores range calls when a range call SUS record is received.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	N/A
Activates/calls:	PERRCHK, RNGSCHD
IPB ID's used:	None
Routines scheduled:	CKRNG
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERRWD : word for error reporting MIGVN : mile mark given REM : fraction of a mile
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets PV08 bits 12, 13, 14 and uses PV08(2, 3, 4, 5, 6).

Title:	P09A
Source file:	P09A.FR
Description:	This routine scores range and accuracy of ADH message when SUS record received.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERRWD : word for error reporting
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets PV09 bits 3-9 and clears PV00 bit 4.

Title:	P09B
Source file:	P09B.FR
Description:	This routine checks that the "too..." message was not appropriately given.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets PV09(2) if "too" message is given when the condition does not exist.

Title:	P10A
Source file:	P10A.FR
Description:	This routine scores clearance requests to tower when panel driver record for clearance request is received.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PPANEL
Cancelled by:	N/A
Activates/calls:	PERRCHK, RNGSCHD
IPB ID's used:	None
Routines scheduled:	CKCLR, CKCWO
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERRWD : word for error reporting
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bits 1-4 of PV10 and clears bit 5 of PV00.

Title:	P10B
Source file:	P10B.FR
Description:	This routine scores the wind message when SUS record for "wind..." is received.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERRWD : word for error reporting
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bits 5, 6 of PV10.

Title:	P10C
Source file:	P10C.FR
Description:	This routine scores clearance message to pilot when SUS record for "cleared..." is received.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERRWD : word for error reporting
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bits 7, 8 of PV10 and clears PV00, bit 6.

Title:	P10D
Source file:	P10D.FR
Description:	This routine scores R/T of waveoff due to clearance problems when a waveoff phrase SUS record is received.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PWAVE
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bit 11 of PV10. It assumes no-gyro is full stop and that min. fuel always cleared. It also sets up call to PMCAM by setting a bit in PVNEX.

Title:	P11A
Source file:	P11A.FR
Description:	This routine scores the OLT advisory when SUS record for OLT is received.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	N/A
Activates/calls:	TIMSCHD
IPB ID's used:	None
Routines scheduled:	CKFCP
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine clears bit 7 of PV00, schedules CKFCP in 3 sec. and schedules OLTCK via PVNEX(5).

Title:	P12A
Source file:	P12A.FR
Description:	This routine scores rollout instructions when a SUS record for "contact..." is received.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	None
Activates/calls:	TIMSCHD, PERRCHK
IPB ID's used:	None
Routines scheduled:	CKPAT
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bit 2 of PV12, clears bit 8 of PV00, schedules CKPAT 10 seconds hence, and schedules PATCK via PVNEX(7).

Title:	P12B
Source file:	P12B.FR
Description:	This routine is called when SUS record for "on the go" is received. It flags the message and schedules handoff check via PVNEX(9).
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	P12C
Source file:	P12C.FR
Description:	This routine scores termination of handoff procedure when "c/s radar contact" special activity record received.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSPEC
Cancelled by:	N/A
Activates/calls:	PERRCHK, TIMSCHD
IPB ID's used:	None
Routines scheduled:	CKRFR
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bit 12 error in PV12.

Title:	P13A
Source file:	P13A.FR
Description:	This routine sets variables and scores no-gyro warning when a turn SUS record is received after gyros have failed but "no-gyro...." has not been announced.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PTURN
Cancelled by:	N/A
Activates/calls:	PERRCHK, RNGSCHD
IPB ID's used:	None
Routines scheduled:	CKNGA
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bit 1 of PV13.

Title:	P13B
Source file:	P13B.FR
Description:	This routine scores the no-gyro approach advisory to pilot when SUS record for "this is a no-gyro approach" is received.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	None
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERRWD : word for error reporting
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bits 1, 3 of PV13, clears PV00 bit 11, and sets bit 12 in PV00.

Title:	P13C
Source file:	P13C.FR
Description:	This routine scores 1/2 standard turns advisory when SUS record received.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERRWD : word for error reporting
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bit 5, 6 of PV13.

Title:	P14A
Source file:	P14A.FR
Description:	This routine scores direction of no-gyro heading correction and stop turn omission when a no-gyro turn record is received.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS, PTURN
Cancelled by:	N/A
Activates/calls:	PERRCHK, DIRT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bits 1, 2 of PV14. It uses PV14(1, 2, 4) as counters.

Title:	P14B
Source file:	P14B.FR
Description:	This routine decrements stop-turn error counter when "stop turn" SUS record is received.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine decrements PV14(2).

Title:	P14SCH
Source file:	P14SCH.FR
Description:	If on no-gyro, this routine schedules CKHDCOR in 20 seconds to check for no-gyro heading correction.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSPEC
Cancelled by:	None
Activates/calls:	TIMSCHD
IPB ID's used:	None
Routines scheduled:	CKHDCOR
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine uses PV14(4).

Title:	P15A
Source file:	P15A.FR
Description:	This routine scores the radar contact lost waveoff when SUS record received.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine clears bit 1 of PV15 and WOCK via PVNEX(8).

Title:	P15BC
Source file:	P15BC.FR
Description:	This routine scores R/T of minimum separation waveoff.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PWAVE
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bit 6 of PV15, clears bit 4 of PV15, and sets up call to PMCAM via PVNEX.

Title:	P15SCH
Source file:	P15SCH.FR
Description:	This routine sets up for minimum separation and radar contact lost waveoff.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSPEC
Cancelled by:	None
Activates/calls:	TIMSCHD
IPB ID's used:	None
Routines scheduled:	None
Cancels:	CKWO
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bits 1, 4 of PV15.

Title:	P16
Source file:	P16.FR
Description:	This routine handles SUS record for "low altitude alert."
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bit 1 of PV16, and clears bit 1 of PV16.

Title:	P17A
Source file:	P17A.FR
Description:	This routine scores the transmission break.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TIMCAL (by P17SCH)
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	P17B
Source file:	P17B.FR
Description:	This routine counts the number of trans. breaks given after the "do not ack..." and prior to 1 mile.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PPANEL
Cancelled by:	None
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	N/A
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	P17SCH
Source file:	P17SCH
Description:	This routine schedules P17A in 3 seconds after SUS record for "over" is received.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	N/A
Activates/calls:	TIMSCHD
IPB ID's used:	None
Routines scheduled:	P17A
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	P18
Source file:	P18.FR
Description:	This routine takes down data for transmission rate check.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PPANEL
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine uses PV18(1, 2, 3).

Title:	P19A
Source file:	P19A.FR
Description:	This routine checks the servoing procedure for the alignment check.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	IPBIN1
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	P19B
Source file:	P19B.FR
Description:	This routine checks the accuracy of alignment requests.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SC19
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	I : loop index BIT : bit number
Files created/changed:	None
Files referenced:	None
Notes:	If alignment was needed but he didn't check the correct display, no credit. If he saw that alignment was needed on one display and didn't bother to check the other, full credit.

Title:	PATCK
Source file:	PATCK.FR
Description:	This routine scores rollout notification to pattern controller when SUS record following "contact tower..." is received.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERRWD : word for error reporting
Files created/changed:	None
Files referenced:	None
Notes:	This routine sets bits 3, 4 of PV12.

Title:	PERRCHK
Source file:	PERRCHK.FR
Description:	This records errors in ER (or PER) or explains them to student depending on the phase. If it is Phase 2, it kills the run afterwards. If it is Phase 3, it updates the PV word with the new error.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	P**, CK-, AF-
Cancelled by:	N/A
Activates/calls:	EXPLAIN, ERINDEX, P2FRZ, RDERR
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVPHZ
Input arguments:	TIME : time of error PVNB : number of PV** ERRWD : bits representing errors detected PVWD : PV** to be updated
Output arguments:	None
Local variables:	ORECORD : record for output to NCERR INDEX : temp IER : error argument I : loop control
Files created/changed:	NCERR
Files referenced:	None
Notes:	None

Title:	PEXCAM
Source file:	PEXCAM.FR
Description:	This routine scores the continuation of a waveoff message after "tower clearance cancelled/not received" is spoken.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERRWD : bit holder
Files created/changed:	None
Files referenced:	None
Notes:	Scheduled by P10D through setting bit 1 of PVNEX. Sets bits 10, 11 and 12 of PV10.

Title:	PHOSCH
Source file:	PHOSCH.FR
Description:	This routine schedules a handoff check under proper conditions.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	WOCK, PWAVE, RNGCAL (by PI12)
Cancelled by:	None
Activates/calls:	TIMSCHD
IPB ID's used:	None
Routines scheduled:	CKHO
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PI00
Source file:	PI00.FR
Description:	This routine initializes PMS universals, PV00 bits and routine triggers.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PMINT
Cancelled by:	None
Activates/calls:	PLACE
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	"Begin descent" and "do not acknowledge" and "this is a no-gyro approach" are recorded regardless of PVN status. Also waveoffs are always recorded.

Title:	PI01
Source file:	PI01.FR
Description:	This is initialization for PV01.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PMINT
Cancelled by:	N/A
Activates/calls:	PLACE
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PI02
Source file:	PI02.FR
Description:	This is initialization for PV02.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PMINT
Cancelled by:	None
Activates/calls:	PLACE
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PI03
Source file:	PI03.FR
Description:	This is initialization for PV03.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PMINT
Cancelled by:	N/A
Activates/calls:	PLACE
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PI04
Source file:	PI04.FR
Description:	This is initialization for PV04.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PMINT
Cancelled by:	N/A
Activates/calls:	PLACE
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PI05
Source file:	PI05.FR
Description:	This is initialization for PV05.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PMINT
Cancelled by:	N/A
Activates/calls:	PLACE
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PI06
Source file:	PI06.FR
Description:	This is initialization for PV06.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PMINT
Cancelled by:	N/A
Activates/calls:	PLACE
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PI07
Source file:	PI07.FR
Description:	This is initialization for PV07.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PMINT
Cancelled by:	None
Activates/calls:	PLACE
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PI08
Source file:	PI08.FR
Description:	This is initialization for PV08.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PMINT
Cancelled by:	N/A
Activates/calls:	PLACE, RNGSCHD
IPB ID's used:	None
Routines scheduled:	CKRNG
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PI09
Source file:	PI09.FR
Description:	This is initialization for PV09.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PMINT
Cancelled by:	N/A
Activates/calls:	PLACE, RNGSCHD
IPB ID's used:	None
Routines scheduled:	CKADH
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PI10
Source file:	PI10.FR
Description:	This is initialization for PV10.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PMINT
Cancelled by:	N/A
Activates/calls:	PLACE, RNGSCHD
IPB ID's used:	None
Routines scheduled:	CKPCLR
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PI11
Source file:	PI11.FR
Description:	This is initialization for PV11.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PMINT
Cancelled by:	N/A
Activates/calls:	PLACE
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PI12
Source file:	PI12.FR
Description:	This is initialization for PV12.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PMINT
Cancelled by:	N/A
Activates/calls:	PLACE, RNGSCHD
IPB ID's used:	None
Routines scheduled:	PHOSCH
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PI13
Source file:	PI13.FR
Description:	This is initialization for PV13.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PMINT
Cancelled by:	N/A
Activates/calls:	PLACE
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PI14
Source file:	PI14.FR
Description:	This is initialization for PV14.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PMINT
Cancelled by:	N/A
Activates/calls:	PLACE
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PI15
Source file:	PI15.FR
Description:	This is initialization for PV15.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PMINT
Cancelled by:	N/A
Activates/calls:	PLACE
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PI16
Source file:	PI16.FR
Description:	This is initialization for PV16.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PMINT
Cancelled by:	N/A
Activates/calls:	PLACE
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PI17
Source file:	PI17.FR
Description:	This is initialization for PV17.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PMINT
Cancelled by:	None
Activates/calls:	PLACE,RNGSCHD
IPB ID's used:	None
Routines scheduled:	CKTB
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PI18
Source file:	PI18.FR
Description:	This is initialization for PV18.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PMINT
Cancelled by:	N/A
Activates/calls:	PLACE,RNGSCHD
IPB ID's used:	None
Routines scheduled:	CKP18
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PI19
Source file:	PI19.FR
Description:	This routine initializes PV19, the alignment check performance measurement variable.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	DEMO
Cancelled by:	N/A
Activates/calls:	IPBOUT1
IPB ID's used:	IDSERVO
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PMCAM
Source file:	PMCAM.FR
Description:	This routine assures correct completion of phrases for "climb and maintain....".
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	N/A
Activates/calls:	PERRCHK
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERRWD : word for error reporting
Files created/changed:	None
Files referenced:	None
Notes:	This routine is used in conjunction with PV10, PV15. It sets bits 3, 6, 9, 12 of PV15. (presently only 3, 12) and bit 11 of PV10. It is scheduled by P10D, WOCK, P15BC.

Title:	PMCLR
Source file:	PMCLR.SR
Description:	This routine initializes a given addressable block to a given value.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	SUSEND, PMINT
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	ARG0 : starting address of area to be initialized ARG1 : number of locations to be initialized ARG2 : initialization value
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PMINT
Source file:	PMINT.FR
Description:	This is the PMS initialization routine. It calls individual initialization routines for those performance variables being scored for a particular run.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PZEC, PB23SUB
Cancelled by:	N/A
Activates/calls:	PMCLR, PI**(**=00-18)
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PMOLT
Source file:	PMOLT.FR
Description:	This routine clears out PV00 bits intended only for waveoffs, low approaches and touch-and-gos.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSPEC
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PMS
Source file:	PMS.FR
Description:	This is the PMS executive. Its sole purpose is to transfer control to the subroutine which processes the encountered student activity.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PZEC, RZEC
Cancelled by:	N/A
Activates/calls:	PSUS, PPANEL, PSPCH, PSPEC
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PMSCHD
Source file:	PMSCHD.FR
Description:	This routine schedules PMS related routines at run time for generation of special activity records.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	MODELINIT
Cancelled by:	None
Activates/calls:	RNGSCHD
IPB ID's used:	None
Routines scheduled:	MILER, MARKIT
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	I : loop index
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PMWAV
Source file:	PMWAV
Description:	This routine cleans up PV00 if waveoff, low approach, or touch-and-go is executed.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSPEC
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	This routine clears bits 6-8 of PV00.

Title:	PPANEL
Source file:	PPANEL
Description:	This routine processes student panel changes for performance measurement.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PMS
Cancelled by:	None
Activates/calls:	TIMSCHD P10A, P17B, P18, PANLOG
IPB ID's used:	None
Routines scheduled:	CKK5, CKCWO, CKKS
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	Present encoding assumes more than one change may occur. Also it assumes that actions taken for each change are independent of all other changes.

Title:	PSPCH
Source file:	PSPCH
Description:	This routine processes automated voice records for PMS.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PMS
Cancelled by:	N/A
Activates/calls:	EXEC
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	EXEC calls a routine indexed by the speech message received.

Title:	PSPEC
Source file:	PSPEC.FR
Description:	This routine handles special event records for performance measurement.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PMS
Cancelled by:	N/A
Activates/calls:	EXEC, TIMSCHD, P05SCH, P14SCH, PMOLT.
IPB ID's used:	None
Routines scheduled:	CKCN, CKCHK, CKLAA, CKACK, CKOLT
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PST1
Source file:	PST1.FR
Description:	This routine handles SUS student activity records when PMS state is 1.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	None
Activates/calls:	EXEC
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	When "wheels..." is given as radio check, it is also graded by PSUS.

Title:	PSUS
Source file:	PSUS.FR
Description:	This subroutine processes student speech input records for PMS.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PMS
Cancelled by:	N/A
Activates/calls:	PST1, EXEC, PMCAM
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	PVEXE routines are listed in PMVC.CO. PVEXE(I) is called for PVNEX, bit I. PVSUB routines are also listed in PMVC.CO.

Title:	PTURN
Source file:	PTURN.FR
Description:	PSUS turn phrases are handled here.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	None
Activates/calls:	EXEC
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PWAVE
Source file:	PWAVE.FR
Description:	PSUS waveoff phrases are handled here.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	N/A
Activates/calls:	EXEC
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PZEC
Source file:	PZEC.FR
Description:	This is the executive for the grading phase. It reads blocks from RPLACT and calls RDACT to fill words in SPACT and call PMS.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	P3TRM, MODIFY
Cancelled by:	N/A
Activates/calls:	SCORE, PMS, PMINT, RNGCAL, TIMCAL, RDACT, RTINIT, RDERR
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	IER : error word PIER : RDACT error word
Files created/changed:	None
Files referenced:	RPLACT
Notes:	None

Title:	SC1214
Source file:	SC1214.FR
Description:	This scores PV12-PV14.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SCORE
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	T : temp to hold score I : loop control
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SC1518
Source file:	SC1518.FR
Description:	This scores PV15-PV18.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SCORE
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	T : temp to hold score I : loop control
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SC19
Source file:	SC19.FR
Description:	This routine scores the alignment checking procedure and gives feedback to the student in case of error.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RTZEC
Cancelled by:	N/A
Activates/calls:	P19B, FB19, IPBOUT1, RDERR, GETNEXT
IPB ID's used:	IDSERVO, IDCRT, IDFF
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	IBIT : loop index CTR : timer NEXT : subroutine argument
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SC35
Source file:	SC35.FR
Description:	This scores PV03-PV05.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SCORE
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	T : temp to hold score I : loop control PTR : for accessing words in loop
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SC68
Source file:	SC68.FR
Description:	This scores PV06-PV08
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SCORE
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	T : temp to hold score
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SC911
Source file:	SC911.FR
Description:	This scores PV09-PV11.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	SCORE
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	T : temp to hold score I : loop control
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SCORE
Source file:	SCORE.FR
Description:	This scores PV01 and PV02 and calls routines to score the rest of the PVs.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PZEC
Cancelled by:	N/A
Activates/calls:	SC35, SC58, SC911, SC1214, SC1518
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	T : temp to hold score I : loop control
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	WOCK
Source file:	WOCK.FR
Description:	This routine scores R/T of ADH and radar contact lost waveoffs.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PSUS
Cancelled by:	None
Activates/calls:	PERRCHK, PHOSCH
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERRWD : word for error reporting
Files created/changed:	None
Files referenced:	None
Notes:	This sets bits 3, 11, 12 of PV15, schedules itself via PVNEX(8). It is also scheduled by P15A and DHCK . It schedules PMCAM for "climb...." through PVNEX and resets bit 10 of PV15.

INTER-PROCESSOR BUS COMMUNICATIONS

Title:	GOOF1
Source file:	GOOF1.FR
Description:	This routine writes out error messages for the IPB or CPU 1.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	IPBOUT1
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	ICODE : error message = 1 : illegal task ID 2 : illegal # of args. 3 : read error on task ID : task ID
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	IPBIN1
Source file:	IPBIN1
Description:	This routine receives and stores arguments from the IPB on CPU 1.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	SYSINIT
Cancelled by:	PZERR, DIE
Activates/calls:	TASKOUT, TSKERRDLY1, P19A
IPB ID's used:	None
Routines scheduled:	None
Cancels:	Kills tasks at CPU 2 request
Mailboxes used:	None
Events referenced:	Wakes tasks at CPU 2 request
Input arguments:	None
Output arguments:	None
Local variables:	IDX : next block to be processed in IPARGS IDX1 : pointer into task ID array
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	IPBOUT1
Source file:	IPBOUT1.SR
Description:	This routine organizes and sends arguments across the IPB on CPU 1.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	ACTIVITY, APENIT, CRSTUFE, DEMO, DESCRPROB, DIE, DIGIN, DWAIT, F1ACINIT, GETANS, INITRT, KREPLAY, KSTUD, KTEACH, LEVEL1, MENU, OVERRIDE, P1AC, P1DIS, P1END, P1INIT, P1PRM, P1RAD, P1TXT, PZFRZ, P23SUB, P2RUN, PZSCREEN, RADAR, RADOUT, REMSEL, REXPLAIN, RPINITAC, RTZEC, RUNKILL, RUNSTOP, SC19, SGNOFF, SYSINIT, VOICTST, YORN
Cancelled by:	N/A
Activates/calls:	GOOF1
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	Accepts 3 forms of variable length inputs as follows FORM 1 : TASKID : ID of task to be attended to on CPU 2 ARG1...ARGn : arguments to send to task FORM 2 : 0 : indicates an array follows N : number of elements in array TASKID : ID of task ARRAY : array to be sent FORM 3 : -1 : indicates string follows TASKID : ID of task on CPU 2 STRING<15> : string
Output aruments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	Warning! IPBIN2's input buffer for forms 1 and 2 is 10 words long. If you need to send more than 10 words, you must change the input buffer size on side 2 (see IPBSTF.CO).

Title:	TASKOUT
Source file:	TASKOUT.FR
Description:	This routine performs functions dependent on an identification code and arguments received from the IPB input routine.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	IPBIN1
Cancelled by:	Self
Activates/calls:	KPROC, SUS, VSPRES, LEVEL1, TSKERRDLY1
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVPHZ
Input arguments:	TID : task ID IPTSKARG : arguments to be passed to the task
Output arguments:	None
Local variables:	I : loop index IBRANCH : used in KPROC start IER : Fortran error code
Files created/changed:	None
Files referenced:	None
Notes:	None

AD-A087 190 LOGICON INC SAN DIEGO CA TACTICAL AND TRAINING SYSTE--ETC F/G 17/9
GROUND CONTROLLED APPROACH CONTROLLER TRAINING SYSTEM (GCA-CTS)--ETC(U)
JUN 80 G BARBER, J BOLLENBACHER, D BREWTON N61339-77-C-0162

UNCLASSIFIED NAVTRAEQUIPC-77-C-0162-3 NL

6 OF 8

AD
A087190

Title:	GOOF
Source file:	GOOF.FR
Description:	This routine is called if an error is found in a message to be sent across the IPB from CPU 1 to CPU 2.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	IPBOUT2
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	ICODE: error message 1 : illegal task ID 2 : illegal # of args. 3 : read error on task ID: task ID
Output arguments:	None
Local variables:	None
Files created/changed:	BUGS2
Files referenced:	None
Notes:	None

Title:	IPBIN2
Source file:	IPBIN2.FR
Description:	This routine receives arguments from across the IPB, stores them, and starts routines to act upon the inputs. It also performs the more simple procedures itself.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	START2
Cancelled by:	DIE
Activates/calls:	TALKOUT, LOOKOUT, LOKFORWARD, TSKERRDLY, CLOK2
IPB ID's used:	None
Routines scheduled:	None
Cancels:	CLOK2
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	N : number of arguments to read TASKID : task ID, destination information IDX : IPFILL temporary storage IDX1 : pointer into array of task IDs BYTES : returned byte count IER : error code returned by Fortran
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	IPBOUT2
Source file:	IPBOUT2.SR
Description:	This routine organizes and sends arguments to CPU 1 across the IPB from CPU 2. It can handle arrays, strings, and lists of arguments.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	HELLO, CKCMN, INIT2RT, PLATEXT, PRESENT, SAID, SKBRD, SKPRO, SPEECH, STOVERRIDE, STUDSTATS, VDCOFF, VDCON
Cancelled by:	N/A
Activates/calls:	GOOF
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	Accepts 3 forms of variable length inputs as follows FORM 1: TASKID : ID of task to be attended to on CPU 2 ARG1...ARGN : arguments to send to task FORM 2: 0 : indicates an array follows N : number of elements in array TASKID : ID of task ARRAY : array to be sent FORM 3: -1 : indicates string follows TASKID : ID of task on CPU 2 STRING<15> : string of characters followed by a carriage return.
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	CKCMN
Source file:	CKCMN.FR
Description:	This background routine monitors the interground communications area for messages from the foreground to CPU 1. It relays these messages over the IPB.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	TUNIT
Cancelled by:	Program termination
Activates/calls:	IPBOUT2
IPB ID's used:	IDRDCHG, IDRADAR
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	IRY : array holding foreground message TID : IDRDCHG MSG : new display baseline (RDCHG) TID1 : IDRADAR MSGSV1 : servo position (RDSVAZ) MSGSV2 : servo position (RDSVEL)
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	FREETOWRCMN
Source file:	FREETOWRCMN.FR
Description:	This function protects the interground communications area in CPU 2 from being overwritten until the foreground has processed the message.
Classification:	Logical function
Period:	N/A
Language:	F
Activated/called by:	LOKFORWARD
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	IDUM : dummy argument
Output arguments:	FREETOWRCMN
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	LOKFORWARD
Source file:	LOKFORWARD.FR
Description:	This routine routes display-related IPB information from CPU 1 to the CPU 2 foreground.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	IPBIN2
Cancelled by:	Self
Activates/calls:	FREETOWRCMN, TSKERRDLY
IPB ID's used:	None
Routines scheduled:	None
Cancels:	Foreground and background
Mailboxes used:	None
Events referenced:	None
Input arguments:	TID : ID of foreground routine which is to receive the message IPLOKARG : message
Output arguments:	TID : set to -1 when information has been copied
Local variables:	IDUM : dummy function argument IER : error code
Files created/changed:	None
Files referenced:	None
Notes:	This routine is responsible, among other things, for bringing the GCA-CTS trainee computer to a graceful return to the CLI when ↑STOP is entered at the instructor station.

Title:	LOOKOUT
Source file:	LOOKOUT.FR
Description:	This routine starts non-interdependent procedures on CPU 2 based on information stored in buffers by IPBIN2, sent across the IPB.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	IPBIN2
Cancelled by:	Self
Activates/calls:	SPEECH, LEVEL, PLATEXT, HEARSAY, TSKERRDLY, STIFLE
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	TID : task ID IPLOKARG : stack array mapped into IPARGS
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	TALKOUT
Source file:	TALKOUT.FR
Description:	This routes information sent by the IPB from CPU 1.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	IPBIN2
Cancelled by:	Self
Activates/calls:	SKPRO, PRESENT, SKBRD, STUDSTATS, STOVERRIDE, HELLO, INIT2RT, TSKERRDLY
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	TID : task ID IPTLKARG : stack array mapped into IPARGS
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

KEYBOARD PROCESSING

Title:	DISPATCH
Source file:	DISPATCH.SR
Description:	This routine serves to transfer control to other routines based on a character or number switch based on DSPA instruction. The dispatch table is set up as follows: word 1 = lower valid value for switch word 2 = higher valid value for switch word 3 = start of table values from lower to upper value Each word in the table shall be either an address for further action or a -1 value.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	KTEACH, KSTUD
Cancelled by:	None
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	SWITCH : index of table where address of routine or label is kept TABLE : table of addresses to jump to, or -1
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	The address of word 1 is actually passed to this routine. If the switch falls outside the bounds, control passes to the next statement after the call. Likewise, if the table value is -1, control passes to the next statement after the call. To set up the addresses in the table in Fortran: TABLE(IND) = IADR ($numberlabel). IADR is a function for returning the address of an argument.

Title:	FILNM
Source file:	FILNM.FR
Description:	FILNM scans an array and replaces all occurrences of nulls, tabs, form feeds, line feeds and carriage returns with Blanks (40_8).
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	HED4
Cancelled by:	None
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	NAME : the array to be processed NWORDS : the number of words to process
Output arguments:	NAME : the processed array
Local variables:	NBYTES : 2 times the number of words I : counter for do loop
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	FOR1
Source file:	FOR1.FR
Description:	This routine outputs type 1 status information for instructor use.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PRNTIT
Cancelled by:	N/A
Activates/calls:	GETDIR
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	CHAN : output channel
Output arguments:	None
Local variables:	LINE : for transferring from FNFORM1 to CHAN BUFF : for reading from NCSR1 and NCSCR TODAY : to pass to DATE TIMAR : to pass to TIME DUMMY : to pass to GETDIR IER : error argument I : loop control RUN : temp for output
Files created/changed:	None
Files referenced:	SCRATCH, FOR1, SR1
Notes:	None

Title:	FOR2
Source file:	FOR2.FR
Description:	This creates type 2 status information for instructor use.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PRNTIT, P3TRM
Cancelled by:	N/A
Activates/calls:	HEAD2, FRDIALOG
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	FOUND : logical for loop control DUMMY : to pass to LIST and GETDIR RUN : to hold record number for the run desired SRECORD : buffer to read from NCSR1 RECORD : buffer to read from NCPV19 NAME : buffer to hold name from NCSR1 TASK : buffer to hold task name from instructor
Files created/changed:	None
Files referenced:	SR1, P3
Notes:	None

Title:	FOR3
Source file:	FOR3.FR
Description:	This creates and outputs type 3 status information for instructor use.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PRNTIT
Cancelled by:	N/A
Activates/calls:	FR301, FR304, FR912, FR3HELP, GRESP, FRDIALOG HEAD3
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	FOUND : logical for loop control I : index J : index RUN : record # of run to be reported DUMMY : to pass to LIST and GETDIR NAME : array for student name RECORD : buffer read from NCPV19 TASK : name of task desired
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	FOR4
Source file:	FOR4.FR
Description:	FOR4 produces the type 4, expanded task summary, available from the PRINTSTATS function of GCA-CTS.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PRNTIT
Cancelled by:	None
Activates/calls:	HED4, WRMES, FRDIALOG
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	IER : ISA error code I : index used to count P3 records
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	FR301
Source file:	FR301.FR
Description:	This handles PV01 for type 3 formatting.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	FOR3
Cancelled by:	N/A
Activates/calls:	FRREST, ERINDEX
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	NCERX
Notes:	None

Title:	FR304
Source file:	FR304.FR
Description:	Handles PV04 for format type 3.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	FOR3
Cancelled by:	N/A
Activates/calls:	FRREST, ERINDEX
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	BUFF : array for transfering from file to file
Files created/changed:	NCLPT
Files referenced:	NCERX
Notes:	None

Title:	FR3HELP
Source file:	FR3HELP.FR
Description:	This is subroutine used by FOR3. It is designed to handle those PV's whose bits are all in one word. It prints a heading and calls FRREST to print the rest.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	FOR3
Cancelled by:	N/A
Activates/calls:	FRREST
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	PVNUM : the PV being printed
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	FR912
Source file:	FR912.FR
Description:	This subroutine is called by FOR3. It sets up the title for PV9-PV12 and reports any errors shown in PV00 for those words.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	FOR3
Cancelled by:	N/A
Activates/calls:	FRREST, ERINDEX
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	FRDIALOG
Source file:	FRDIALOG.FR
Description:	This routine handles the dialog with the instructor regarding printouts.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	FOR2, FOR3, FOR4
Cancelled by:	N/A
Activates/calls:	GRESP, LIST, GETDIR
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	$: error return
Output arguments:	SRECORD : SR1 record which applies TASK : task name
Local variables:	FOUND : record found DONE : end of file ENTRY : temporary for instructor request REC : temporary for instructor request I,J,JCR : loop indicies DUMMY : dummy function IER : error code
Files created/changed:	None
Files referenced:	SR1
Notes:	None

Title:	FRREST
Source file:	FRREST.FR
Description:	Handles reporting of errors for form 3 format for all errors not listed in PV00.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	FR3HELP, FR301, FR304, FR912
Cancelled by:	N/A
Activates/calls:	ERINDEX
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	PVNUM : number of PV being printed PVWD : PV word being printed BIT : the last bit handed for this word
Output arguments:	None
Local variables:	None
Files created/changed:	NCLPT
Files referenced:	ERXFI
Notes:	None

Title:	GAMOD
Source file:	GAMOD.FR
Description:	This gets and modifies records from RPLACT.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	MODIFY
Cancelled by:	N/A
Activates/calls:	GRESP, SUBMODIFY, RPFOR
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	REC : to hold record from RPLACT IER : error argument
Files created/changed:	RPLACT
Files referenced:	RPLACT
Notes:	None

Title:	GETDIR
Source file:	GETDIR.FR
Description:	This sets the student's directory for formatting routines.
Classification:	Function
Period:	None
Language:	F
Activated/called by:	FOR1, FRDIALOG, GPRUN
Cancelled by:	N/A
Activates/calls:	PKNM, GRESP, SCHREAD
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	Returns a logical as its name
Local variables:	NAME : for student's last name FNAME : file name, used only as place holder DISK : to tell instructor where files are
Files created/changed:	None
Files referenced:	Student.IX
Notes:	None

Title:	GPRUN
Source file:	GPRUN.FR
Description:	This routine makes the student directory the default and opens the replay channels to P-run. If either of these cannot be done, it returns false.
Classification:	Function
Period:	None
Language:	F
Activated/called by:	MODIFY, KREPLAY
Cancelled by:	N/A
Activates/calls:	GETDIR, OPRDPHZ
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	DUMMY : just that
Output arguments:	None
Local variables:	ANS : to get char IER : error argument
Files created/changed:	None
Files referenced:	P-run replay files
Notes:	None

Title:	GREAL
Source file:	GREAL.FR
Description:	This routine converts a real number to an integer.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	WRMES
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	REALIN : real input
Output arguments:	INTEGEROUT : integer output
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	GRESP
Source file:	GRESP.FR
Description:	This routine retrieves yes/no responses from the keyboard.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	FOR3, KREPLAY, MODIFY, NEWTE, FRDIALOG, GAMOD, GETDIR, OVERRIDE, PRNTIT, SUBMODIFY
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	RETURN 1 : return address if "yes" entered RETURN 2 : return address if "no" entered RETURN 3 : return address if some other character was entered
Output arguments:	None
Local variables:	ANS : key pressed and received by GCHAR IER : Fortran error code
Files created/changed:	None
Files referenced:	None
Notes:	This routine assumes any spacing prior to the prompt has been done by the caller.

Title:	GRESP2
Source file:	GRESP2.FR
Description:	This routines retrieves yes/no responses from the keyboard.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	HELLO, INIT2RT, STOVERRIDE
Cancelled by:	N/A
Activates/calls:	None
IPB IDs used:	None
Routines Scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	Return 1 : return address if "yes" entered Return 2 : return address if "no" entered Return 3 : return address if some other character was entered
Output arguments:	None
Local variables:	ANS : input from the keyboard IER : Fortran error code
Files created/changed:	None
Files referenced:	None
Notes:	This routine assumes that any spacing prior to the prompt has been done by the caller.

Title:	HEAD2
Source file:	HEAD2.FR
Description:	Prints a heading for FOR2.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	FOR2
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	NAME : student name RECORD : other student info
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	HEAD3
Source file:	HEAD3.FR
Description:	This writes the heading for type 3 formatted output.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	FOR3
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	NAME : 24 word array with student's name RECORD : 24 word task summary block RUN : the run number requested
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	HED4
Source file:	HED4.FR
Description:	Prints a header for a Type 4 printout.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	FOR4
Cancelled by:	None
Activates/calls:	FILNM
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	NAME : name of student RECORD : SR1 record MINRUNS : GZMNR MAXRUNS : GZNR
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	HELLO
Source file:	HELLO.FR
Description:	This routine processes student sign-on (HELLO) request by receiving the trainee's name, opening his file, flagging an error if no routine is found and notifying CPU 1 when finished.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TALKOUT
Cancelled by:	N/A
Activates/calls:	PKNM, GRESP2, IPBOUT2, SKBRD
IPB ID's used:	IDSINON, IDKPROC
Routines scheduled:	None
Cancels:	SKBRD
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	IER : Fortran error I : loop index Count : RDLIN byte count Name : array for student name
Files created/changed:	None
Files referenced:	DP2F:<name>.IX, the student's index file
Notes:	None

Title:	IADR
Source file:	IADR.SR
Description:	A function which produces the address of a Fortran variable, array, or common.
Classification:	Function
Period:	None
Language:	A
Activated/called by:	KSTUD, KTEACH
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	$ADDRESSLABEL : label of a statement
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	IADR can be used in subroutine calls implicitly.

Title:	IGOODKY
Source file:	IGOODKY.FR
Description:	This routine checks for legal key based on instructor menu.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	KTEACH
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	KEY : key pressed WHO : keyboard from which the key came
Output arguments:	GOOD : T if key was good KEY : index of good key in KBIN
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	IKBRD
Source file:	IKBRD.FR
Description:	This is the keyboard listening task on the instructor side.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	SYSINIT, KREPLAY, NEWTE, STSK
Cancelled by:	KTEACH, MODIFY, OVERRIDE, PRNTIT, NEWTE, KREPLAY
Activates/calls:	KPROC, TSKERRDLY
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVKEY
Input arguments:	None
Output arguments:	None
Local variables:	KEY : key pressed on instructor keyboard
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	INITRT
Source file:	INITRT.FR
Description:	This routine initiates the formation of VRPS at instructor keyboard request.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	STSK
Cancelled by:	Self, STSK
Activates/calls:	IPBOUT1
IPB ID's used:	IDCRT, IDINIT2RT
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVKYST
Input arguments:	None
Output arguments:	None
Local variables:	IER : Fortran error code
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	INIT2RT
Source file:	INIT2RT.FR
Description:	This routine initiates the formation of VRPS at student keyboard request.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	TALKOUT
Cancelled by:	Self
Activates/calls:	IPBOUT2, SKBRD, GRESP2, SPEECH, PRESENT, TSKERRDLY
IPB ID's used:	IDAWAKE
Routines scheduled:	None
Cancels:	SKBRD
Mailboxes used:	None
Events referenced:	EVPHZ, EVKYST
Input arguments:	None
Output arguments:	None
Local variables:	PHRASE : phrase number to be trained NUM : number of repeats IER : Fortran error code I : loop index SPARY : array for speech arguments PRARY : array for present arguments DEVICE : prompting device: 1 : $VRO 2 : $TTO
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	IVT
Source file:	IVT.FR
Description:	This routine starts common processing for INIT V/T keyboard requests.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	KSTUD, KTEACH
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	KPROC
Source file:	KPROC.FR
Description:	This routine routes keys to the instructor or student keyboard processing.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	IKBRD, TASKOUT
Cancelled by:	Self
Activates/calls:	KTEACH, KSTUD, MENU
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVPRC
Input arguments:	KEY : input from keyboard WHO : who sent it 0 : student 1 : instructor 2 : internal routine
Output arguments:	KEY : when called by taskout, indicates completion of processing
Local variables:	IKEY : temporary for KEY IWHO : temporary for WHO
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	KREPLAY
Source file:	KREPLAY
Description:	This routine allows instructor to replay a P-run.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TZEC
Cancelled by:	N/A
Activates/calls:	GRESP, REPLAY, GPRUN, IPBOUT1, GETNEXT, YORN, IKBRD, RLDIR
IPB ID's used:	IDCRT, IDFF
Routines scheduled:	None
Cancels:	IKBRD
Mailboxes used:	None
Events referenced:	EVPHZ
Input arguments:	None
Output arguments:	None
Local variables:	DUMMY : to pass to GPRUN
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	KSTUD
Source file:	KSTUD.FR
Description:	This routine processes inputs from the student keyboard.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	KPROC
Cancelled by:	Self
Activates/calls:	IPBOUT1, DISPATCH, SGOODKY, KTEACH, IADR, MENU, IVT, SVT
IPB ID's used:	IDMENU, IDSTUDSTATS, IDHELLO, IDSERVO, IDSKPRO
Routines scheduled:	None
Cancels:	Self
Mailboxes used:	None
Events referenced:	EVKEY
Input arguments:	KEY : key pressed on student side
Output arguments:	None
Local variables:	MSG : MSG to send across the IPB GOOD : T if key is good KEYTEMP : temporary key storage ITM : time array IER : Fortran error code
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	KTEACH
Source file:	KTEACH.FR
Description:	This routine processes inputs from the instructor keyboard.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	KPROC, KSTUD
Cancelled by:	Self
Activates/calls:	IGOODKY, DISPATCH, IPBOUT1, SPFR, SPGO, CLOK, IVT, SVT, IADR, OEBL, MENU, SHFSTOP
IPB ID's used:	IDSKPRO, IDTIME, IDMENU, IDDIE
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVPHZ, EVKEY
Input arguments:	KEY : key pressed WHO : 0 : student keyboard 1 : instructor keyboard 2: internal request
Output arguments:	MSG : IPB output message for SKPRO if called by KSTUD
Local variables:	GOOD : true if key is good MSG : for IPB for further processing on student side KEYTMP : key temporary storage ICT : used for menu output I,J : loop counters LATONCE : true if terminate request is immediate
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	LIST
Source file:	LIST.FR
Description:	This prints a list of Phase 3 runs for the instructor.
Classification:	Function
Period:	None
Language:	F
Activated/called by:	FRDIALOG
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	Returns T if list was possible
Local variables:	BUFF : for reading from NCSR1 TASK : to hold name input by instructor NAME : to hold first record from NCSR1 TODAY : to pass to DATE IER : error argument I : loop control FOUND : boolean loop control
Files created/changed:	None
Files referenced:	NCSR1
Notes:	NCSR1 is assumed to be open.

Title:	MENU
Source file:	MENU.FR
Description:	This routine determines which menu bit is legal, based on phase, etc.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	KTEACH, KSTUD, KPROC, P1VDC, PZREQ, SGNOFF, VOICTST
Cancelled by:	N/A
Activates/calls:	IPBOUT1
IPB ID's used:	IDMENU
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	ARG : 0 : set menu unless special request received 1 : reset menu after special request
Output arguments:	None
Local variables:	ITMP : temporary menu storage
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	MODIFY
Source file:	MODIFY.FR
Description:	This allows the instructor to modify a trainee's P-run activity file.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	STSK
Cancelled by:	N/A
Activates/calls:	GAMOD, PZEC, GRESP, GPRUN, RPFOR, PANIT, RLDIR, PUTSCORES
IPB ID's used:	None
Routines scheduled:	None
Cancels:	IKBRD
Mailboxes used:	None
Events referenced:	EVZEC
Input arguments:	None
Output arguments:	None
Local variables:	DUMMY : to pass to GPRUN IER : error argument
Files created/changed:	FNPERR
Files referenced:	None
Notes:	None

Title:	NEWTE
Source file:	NEWTE.FR
Description:	This routine creates files for new trainee, including scratch files, digitized speech files and performance files.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	STSK
Cancelled by:	Self, STSK
Activates/calls:	GRESP, PKNM, IKBRD, SCHINIT
IPB ID's used:	None
Routines scheduled:	None
Cancels:	IKBRD
Mailboxes used:	None
Events referenced:	EVZEC
Input arguments:	None
Output arguments:	None
Local variables:	NAME : trainee's last name FNAME : trainee's first name TNAME : trainee's first name, padded with nulls IDENT : trainee's serial number DISK : disk identification INDEX : index file name I : loop index IRESP : instructor response IER : Fortran error code
Files created/changed:	SR1, P3, SUM, SCRATCH, PV19, FORM1, CIDVFILE
Files referenced:	None
Notes:	None

Title:	OEBL
Source file:	OEBL.SR
Description:	This routine complements the state of console interrupt enable status. There are identical versions in CPU 1 and CPU 2.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	KTEACH, SKBRD, SYSINIT
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	KBTYPE : current state of CTRL/C: =0 : CTRL/C enabled, =1 : CTRL/C disabled
Output arguments:	KBTYPE : new state of CTRL/C, as above
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	OKRT
Source file:	OKRT.FR
Description:	OKRT is used in the New R/T function to ask the user which phrases he wants to retrain and the number of repeats to retrain. These entries are then checked for validity.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	INIT2RT
Cancelled by:	None
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVKEY
Input arguments:	OLD PHRASES : logical array that keeps track of phrases trained
Output arguments:	PHRASE : the phrase to retrain NUM : the number of repeats to train IER : set to 1 id everything is ok, else 0 to indicate end of training
Local variables:	MAXNUM : the maximum allowable number of repeats
Files created/changed:	None
Files referenced:	None
Notes:	The instructor may train all repeats of a particular phrase. The student may only train up to 1/2 of the total number of repeats used.

Title:	OPRDPHZ
Source file:	OPRDPHZ.FR
Description:	This routine opens the P-run problem file for MODIFY. It saves the file pointer if the file is already open.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	GPRUN
Cancelled by:	None
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	FOUND : indicates whether the P-run card was found
Local variables:	I : loop index IER : error POSITION : saves old file position LINE : holds data read from file
Files created/changed:	None
Files referenced:	PRUN (a link to T06$00.04)
Notes:	None

Title:	OVERRIDE
Source file:	OVERRIDE.FR
Description:	This routine handles override keyboard requests.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	STSK
Cancelled by:	STSK, Self
Activates/calls:	IPBOUT1, GRESP
IPB ID's used:	IDSTOVER, IDCRT
Routines scheduled:	None
Cancels:	IKBRD
Mailboxes used:	None
Events referenced:	EVPHZ, EVZEC
Input arguments:	None
Output arguments:	None
Local variables:	ENTRY : instructor selection I : loop index IER : Fortran error code NEWLEVEL : decoded level of new task NEWPHASE : decoded phase of new task NEWTSK : new task name OLDLEVEL : decoded level of last task OLDTASK : decoded task of last task OLDTSK : name of last task
Files created/changed:	None
Files referenced:	task file, SUM
Notes:	This routine assumes the default directory is DP2F, and furthermore that the override task should be found there. It also assumes that the task names are all of the form T**$**.**, where the ** decodes to level, task and phase respectively.

Title:	PKNM
Source file:	PKNM.SR
Description:	This simple routine constructs the index file name based upon the student's name.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	NEWTE, HELLO, GETDIR
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	ARRAY : array containing student name
Output arguments:	ARRAY : array containing index file name
Local variables:	TM1 : holds address of array element being processed
Files created/changed:	None
Files referenced:	None
Notes:	This routine is very simple minded. It assumes the name array ends with one space. It further assumes that the input array is large enough to hold the index file name with its extension. It will overwrite core if these conditions are not met.

Title:	PRHELP
Source file:	PRHELP.FR
Description:	This helps RPFOR print replay report. It prints record sent to it from RPLACT.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RPFOR
Cancelled by:	N/A
Activates/calls:	PRSUS
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	REC : a GLIB or SUS record from RPLACT RPTR : record pointer into RPLACT
Output arguments:	RPTR : updates this if passed a SUS record
Local variables:	MILE : to convert to real for printing BUFF : to hold a record from NCPH SPKR : to hold speaker mnemonic TIME : a temp PTR : a pointer into the record passed
Files created/changed:	None
Files referenced:	NCPH
Notes:	None

Title:	PRNTIT
Source file:	PRNTIT
Description:	This asks instructor which type of printout he wants and activates the appropriate routine.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	STSK
Cancelled by:	Self
Activates/calls:	FOR1, FOR2, FOR3, FOR4, RLDIR, RDERR, GRESP
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVZEC
Input arguments:	None
Output arguments:	None
Local variables:	NUM : type of printout CHN1 : a type 1 output channel
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PRSUS
Source file:	PRSUS.FR
Description:	This prints a SUS record to the channel passed to it.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PRHELP, REXPLAIN, SUBMODIFY
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	CHAN : channel for output PHRASE : record number for phrase in "VOTEXT" CS : call sign number (1-4) HDG : recognized heading SPEED : recognized speed CORRECTION : correction applied?
Output arguments:	None
Local variables:	CALLSIGN : for holding record from "VOTEXT" BUFF : for holding record from "VOTEXT"
Files created/changed:	CHAN
Files referenced:	RPPACT, RPLACT, VOTEXT
Notes:	This routine prints the heading if <>-1, i.e., it doesn't check if it is a 'heading' phrase before printing the number.

Title:	PUTSCORES
Source file:	PUTSCORES.FR
Description:	This routine puts the MODIFYed scores resulting from the correction of the P-run file into the trainee files.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	MODIFY
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	P3, SUM
Files referenced:	None
Notes:	None

Title:	RLDIR
Source file:	RLDIR.FR
Description:	This routine closes the files opened by GETDIR and GPRUN.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	KREPLAY, MODIFY, PRNTIT
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	ARG : indicates which files are open
Output arguments:	None
Local variables:	IER : error code
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	RPFOR
Source file:	RPFOR.FR
Description:	This prints the replay report after a run.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	GAMOD, P3TRM, MODIFY
Cancelled by:	N/A
Activates/calls:	RPHEAD, PRHELP, RPKEY
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	ERREC : to hold a record from ERRFI RPREC : to hold a record from RPLACT BUFF : to hold a record from ERXFI ERPTR : record pointer into ERRFI RPPTR : record pointer into RPLACT ERIER : error argument RPIER : error argument NEXTSTOP : time to report next error
Files created/changed:	None
Files referenced:	RPLACT, ERRFI, ERXFI
Notes:	None

Title:	RPHEAD
Source file:	RPHEAD.FR
Description:	This prints a heading for the replay report.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RPFOR
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	BUFF : to hold record from NCSR1 TIMAR : to pass to TIME TODAY : to pass to DATE I : loop control IER : error argument
Files created/changed:	None
Files referenced:	NCSR1
Notes:	None

Title:	RPKEY
Source file:	RPKEY.FR
Description:	This routine reports mike key state changes on the P-run printout.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	RPFOR
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	RPREC : panel change record RPPTR : record number of RPREC MIKE : previous mike key state
Output arguments:	MIKE : current mike key state
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SGNOFF
Source file:	SGNOFF.FR
Description:	This routine updates student records to indicate a signoff. It closes all student files and indicates time of completion within these files.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	TZEC
Cancelled by:	N/A
Activates/calls:	IPBOUT1, MENU, SCHWRITE
IPB ID's used:	IDSKPRO
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVPHZ
Input arguments:	None
Output arguments:	NEXT : indicates that demc is the next task to start
Local variables:	ITIME : time array IDATE : date array LCBUFF : record read from performance file
Files created/changed:	None
Files referenced:	Closes all trainee files
Notes:	None

Title:	SGOODKY
Source file:	SGOODKY.FR
Description:	This routine determines valid keys based on bit set by MENU.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	KSTUD
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	KEY : key pressed
Output arguments:	KEY : index of good key in KBST GOOD : T if key pressed
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SHFSTOP
Source file:	SHFSTOP.FR
Description:	This routine handles "terminate GCA-CTS" requests.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	KTEACH
Cancelled by:	Self
Activates/calls:	DIE
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	LATONCE : true if stop is to be at once; else false
Local variables:	RESPONSE : instructor response IER : Fortran error code
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SINON
Source file:	SINON.FR
Description:	This routine initializes CPU 1 for a student who has just signed on.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	PZDEMO
Cancelled by:	Self
Activates/calls:	DIE
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	IER : Fortran error code ITMP : string holder
Files created/changed:	None
Files referenced:	SCRATCH, SR1
Notes:	None

Title:	SKBRD
Source file:	SKBRD.FR
Description:	This is the keyboard listening task, on the student side. It does validity checking.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	START2, HELLO, INIT2RT, STOVERRIDE, TALKOUT
Cancelled by:	HELLO, INIT2RT, STOVERRIDE
Activates/calls:	IPBOUT2, OEBL
IPB ID's used:	IDKPROC
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVKEY
Input arguments:	None
Output arguments:	None
Local variables:	KEY : key pressed
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SKPRO
Source file:	SKPRO.FR
Description:	This routine processes menu requests and other keyboard processing on the trainee side.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	TALKOUT
Cancelled by:	Self
Activates/calls:	VIPON, VIPOFF, IPBOUT2
IPB ID's used:	IDAWAKE
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVPHZ
Input arguments:	MSG : switch used to handle further processing
Output arguments:	None
Local variables:	I,J : loop indices ICT : counter for menu display IER : Fortran error code
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	STHELP
Source file:	STHELP.FR
Description:	This helps CRSTUFE write stuff to NCSTFE.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	CRSTUFE
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	J : PV number to be printed
Output arguments:	None
Local variables:	None
Files created/changed:	STUFE
Files referenced:	None
Notes:	None

Title:	STOVERRIDE
Source file:	STOVERRIDE.FR
Description:	This routine handles override keyboard requests on the student keyboard.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	TALKOUT
Cancelled by:	Self
Activates/calls:	IPBOUT2, SKBRD, GRESP
IPB ID's used:	IDOVERRIDE
Routines scheduled:	None
Cancels:	SKBRD
Mailboxes used:	None
Events referenced:	None
Input arguments:	OLD : old task name SW : 1 if present task is not override 2 if present task is override
Output arguments:	None
Local variables:	ENTRY : instructor selection I : loop index IER : Fortran error code NEWLEVEL : decoded level of new task NEWPHASE : decoded phase of new task NEWTASK : decoded task of new problem NEWTSK : new task name OLDLEVEL : decoded level of last task OLDTASK : decoded task of last task OLDTSK : name of last task PRTISOVER : holder for SW STATUS : RSTAT array
Files created/changed:	None
Files referenced:	P3, SUM
Notes:	This routine assumes the default directory is DP2F, and furthermore that the override task should be found there. It also assumes that the task names are all of the form T**$**.**, where the ** decodes to level, task and phase respectively.

Title:	STUDSTATS
Source file:	STUDSTATS.FR
Description:	This routine reads student statistics from STUFE file and puts them on the student CRT.
Classification:	TASK
Period:	None
Language:	F
Activated/called by:	TALKOUT
Cancelled by:	Self
Activates/calls:	IPBOUT2
IPB ID's used:	IDAWAKE
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVTXT
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	STUFE
Notes:	This routine assumes that STUFE is closed.

Title:	SUBMODIFY
Source file:	SUBMODIFY.FR
Description:	This modifies a record for MODIFY. It divides a record into sections like 'heading' and asks if it was reported correctly. If not, it asks what it should be and changes it.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	GAMOD
Cancelled by:	N/A
Activates/calls:	PRSUS, GRESP
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	REC : record to be modified
Output arguments:	REC : record modified
Local variables:	TEMP : for reading
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	SVT
Source file:	SVT.FR
Description:	This routine handles common stop voice test processing.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	KSTUD, KTEACH
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	EVSTP
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	WRMES
Source file:	WRMES.FR
Description:	WRMES prints a Form 4 problem summary including scores and information about the environmental conditions used in that approach.
Classification:	Subroutine
Period:	None
Language:	F
Activated/called by:	FOR4
Cancelled by:	None
Activates/calls:	GREAL
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	CHANNEL : the output channel to use N : the record number in P3 we are accessing REC : the P3 record
Output arguments:	None
Local variables:	ITMP : used to hold the proper character to print for T/F variables ITIME : an array used to convert the time to the proper format
Files created/changed:	None
Files referenced:	None
Notes:	None

TRAINEE AND INSTRUCTOR PANEL ROUTINES

Title:	PANEL
Source file:	PANEL.FR
Description:	This receives an IXMT from the panel interrupt service routine and starts whatever processing is necessary based upon the phase. It also handles the update of the replay file as necessary.
Classification:	Task
Period:	None
Language:	F
Activated/called by:	SYSINIT
Cancelled by:	DIE
Activates/calls:	ACTOUT, TIMSCHD
IPB ID's used:	None
Routines scheduled:	TOWER
Cancels:	None
Mailboxes used:	BXPAN
Events referenced:	EVPNL
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PANIT
Source file:	PINDR.SR
Description:	This routine is called to initialize panel logicals without attaching the panel to the RDOS interrupt structure.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	REPLAY, SCORE, PZ23, LEVEL1, MODIFY, P1INIT, P1PRM, P2RNSTOP, P[illegible], PB23SUB, RUNSTOP
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	ARG [illegible] operation [illegible] panel responses for SCORE and REPLAY
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PANOFF
Source file:	PINDR.SR
Description:	This routine is called to remove the panel from the RDOS interrupt structure and turns off the device and lights.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	DIE
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PANON
Source file:	PINDR.SR
Description:	This subroutine initializes KEYS.CO logicals, attaches the panel to the RDOS interrupt structure, then starts the device.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	SYSINIT
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	None

Title:	PANLOG
Source file:	PANOUT.SR
Description:	This routine is used during scoring to set the logicals as though the activity file entries had been observed by the panel interrupt service routine. Since the driver makes use of the unused DOB bits to record status information, only the DOA and DOB words are needed.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	PPANEL
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	ARG0 : DOA word ARG1 : DOB word
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	The panel must be disabled prior to the use of this routine by CALL PANIT (KYOFF) because otherwise the current state of the panel would affect the logicals.

Title:	PANOUT
Source file:	PANOUT.SR
Description:	This routine is used to change the state of the lights on the trainee panel and the corresponding logicals. It does this by changing the panel output word settings, then starting the device so that the interrupt service routine outputs the new values. This implementation makes it unnecessary for PANOUT to issue an INTDS, and also keeps panel-related output to the activity file centralized.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	CLRBUTX, CLREQ, CLRNC, DESEL, ENDFEED, FEED, GO, IGNORE, MODELINIT, P1PRM, P23SUB, SELBUT, WALOFF, WAVE
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	ARG0 : light to be changed, from KEYS.CO parameters ARG1 : on/off switch, from KEYS.CO parameters
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	Although the alarm can be turned on by itself, it also is turned on and off automatically with the waveoff light. Also, when the amber frequency light is turned off, if the frequency is selected the light turns green and the alarm goes off.

Title:	PIN
Source file:	PINDR.SR
Description:	The panel interrupt service routine services all panel interrupts, builds and outputs the corresponding values to set the lights, etc., and sets the corresponding logical values. It performs .IWAKES when the student voice level changes and when $VRO stops. It sends an .IXMT to the panel, a high level routine which performs activity file updates.
Classification:	Interrupt service
Period:	None
Language:	A
Activated/called by:	Panel interrupt
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	BXPAN
Events referenced:	EVVST, EVVIN, EVVRO
Input arguments:	None
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	The panel initialization (PANIT, PANON) and removal (PANOFF) subroutines are included in PINDR. These are FORTRAN callable assembly language routines.

Title:	REPAN
Source file:	PANOUT.SR
Description:	This routine is used during replay to recreate the panel light displays without deactivating the SUPER/ICS buttons.
Classification:	Subroutine
Period:	None
Language:	A
Activated/called by:	ACTIVITY
Cancelled by:	N/A
Activates/calls:	None
IPB ID's used:	None
Routines scheduled:	None
Cancels:	None
Mailboxes used:	None
Events referenced:	None
Input arguments:	ARG0 : DOA word ARG1 : DOB word
Output arguments:	None
Local variables:	None
Files created/changed:	None
Files referenced:	None
Notes:	The panel must be disabled by CALL PANIT (KYOFF) before this routine is invoked.

APPENDIX B

COMMON VARIABLE AND PARAMETER DEFINITIONS

Much of the communication between GCA-CTS routines will take place through labeled commons. The variable names within a common block will, in general, begin with the same two letters. The naming conventions as well as the common blocks themselves are defined in this appendix. Block names shown with an asterisk are parameter lists only.

Variable Name Begins with	Common Block Name	Page No.
AC	ACFIX	550
BK	BGROUND*	551
CL	CLOCK1	551
CL	CLOCK2	551
CT	CTRLR	552
NC	DEV1	555
NC	DEV2	556
EM	EMERGE	557
EN	ENVIRON	558
ER	ERR	559
EV	EVNT2*	560
EV	EVNTS*	560
FG	FGROUND	561
FN	FIL1	562
FN	FIL2	564
FZ	FZ1	565
FZ	FZEC	565
GZ	GZEC	566
	IDEXEC*	569
ID	IDPRI1*	570
ID	IDPRI2*	576
IP	IPB1STF	579
IP	IPBSTF	580
KB	KBRD	581
KB	KBRD2	584

Variable Name Begins with	Common Block Name	Page No.
KY	KEYS	585
BX	MAIL1	587
BX	MAIL2	587
MN	MENU1	587
MN	MENU2	588
MS	MSTRING	588
	PARM1*	588
	PARM2*	588
PC	PCP	589
PC	PCP1*	591
PD	PDIGT*	592
PF	PFSCR	592
RP	PLAY	593
PT	PLT	594
PS	PMSSUP	597
PV	PMVC	598
PR	PRMPT	603
P3	PZ3CM	604
RD	RDR	606
RD	RDR1*	607
RC	RECKON	608
SB	SBF	609
SS	SHUSH	610
SK	SKED	613
SA	SPACT	614
SS	SPCH	616
SD	SPDGT	617
SM	SPEECHMESS*	618
SV	SRV	619

Variable Name Begins with	Common Block Name	Page No.
SV	SRV1*	620
SU	SUSAY	621
TE	TEXT2	622
TZ	TZC	622
VI	VICOM	622
IN	VINA1	623
VL	VLID	623
VC	VOCIN	623
	VSCO1	624
VS	VSIFP	625
VX	VX*	627
XP	XPOSE	632

*Parameters only

ACFIX.CO, CPU 1 Aircraft Variables.

Common Variable	Type	Description
ACX	R	X-coordinate of current a/c position (ft)
ACXD	R	X-component of current a/c velocity (ft/sec)
ACY	R	Current a/c altitude (ft)
ACYD	R	Current a/c rate of ascent (ft/sec)
ACZ	R	Z-coordinate of current a/c position (ft)
ACZO	R	Initial value of ACZ this apprch (ft)
ACZR	R	Z-component of radar range (ft) (ACZ+3605.07)
ACZD	R	Z-component of current a/c velocity (ft/sec)
ACH	R	Current a/c heading (rad magnetic)
ACHD	R	Current a/c rate of turn (rad/sec) (-lftwrd)
ACAS	R	Current a/c airspeed (ft/sec)
ACGS	R	Current a/c groundspeed (ft/sec)
ACGA	R	Current gust-induced vrtcl acclrtn (ft/sec)**2
ACQMX	R	Zone boundary function coefficient (fmax)
ACQMN	R	Zone boundary function coefficient (fmin)
ACQF	R	Zone-zone oscillation frequency parameter
ACZVF	R	APREX/APRAX airspeed z-component (ft/half-sec)
ACEZN	I	Current a/c elevation zone
ACAZN	I	Current a/c azimuth zone
ACZBF	RA	Displ of zone inner boundary from g/p or course, as pct of target size (by zone #)
ACTYP	I	Aircraft type code
ACCS	I	Aircraft call sign
ACNEW	L	True if hitherto frozen aircraft may now begin approach
ACGYRO	L	True if gyrocompass now operating
ACENG	L	True if all engines not now operating
ACICE	L	True if control surfaces are now iced
ACHYDR	L	True if hydraulic system is not now operating

Parameter Name	Value	Description
ACMFCONV	6076.11	"Nautical miles" to feet conversion

BGROUND.CO, Parameters defining offsets in Interground Communications Area. These values must correspond to FGROUND.CO offsets.

Parameter Name	Value	Description
BKMSG	0	Offset of message code
BKPICUP	1	Offset of PICUP information
BKSERVO	8	Offset of SERVO information
BKIMAGES	11	Offset of IMAGES information
BKIPB	13	Offset of SETIT messages for IPBOUT2
BKSRV	15	Offset of SERVO messages for IPBOUT2
BKSPCL	18	Offset of special messages
BKSTRING	19	Offset of Megatek strings

CLOCK1.CO, CPU 1 Clocks and Timers.

Common Variable	Type	Description
CL50	I	50 minute class period timer
CLTGH	I	Time to go home
CLTICK	I	Half-second user clock
CL100	I	100 msec user clock
CLTG50	I	Time 50% of target appears

CLOCK2.CO, CPU 2 Clock Common.

Common Variable	Type	Description
CLTICK	I	Time in half second ticks from the start of the problem
CL100	I	Time in 100 msec ticks from above time

CTRLR.CO, CPU 1 Model Controller Information.

Common Variable	Type	Description
CTCLR	I	Clearance type
CTHOF	L	True if feeder is to give hand-off
CTGPP	I	Correct glidepath position message
CTGPT	I	Correct glidepath trend message, or -1
CTCRP	I	Correct course position message
CTCRT	I	Correct course trend message, or -1
CTRNG	I	Correct range, including DH and OLT, or -1
CTOTHR	I	Other final controller messages
CTEMERG	I	Emergency waveoff, etc., or -1
CTREL	L	True if feeder is to fail to relinquish radio frequency until requested to do so
CTNGR	L	True if feeder is to give no gyro hand-off
CTFREQ	I	Frequency for this problem 1:270.8:2:318.8
CTSPH	I	Last feeder controller output
CTFDOVR	L	T if feed finished or expert going (equiv to CTEXPTON)
CTON	L	True if controller is giving the demo
CTPHZ	I	Phase of flight (1, 2 or 3)
CTDEV	I	Requested final controller speech device (1:$VRO, 2:CRT, 3:audio)
CTACK	L	True if pilot is still acknowledging controller messages
CTGPOS	L	True if position message on glidepath is legal
CTCPOS	L	True if position message on course is legal
CTHEAD	I	Course heading
CTLSAZN	I	Last zone position of aircraft on azimuth
CTLSEZN	I	Last elevation zone position of aircraft
CTOHDG	I	Old heading
CTACOHDG	R	Old heading plane actually traveled at
CTAG	RA	Legal speed and range for approaching glidepath
CTBGDS	L	T if begin descent is in CTMSG and thereafter
CTATHT	L	T if at decision ht

CTRLR.CO, CPU 1 Model Controller Information (Continued)

Common Variable	Type	Description
CTNGA	IA	No gyro array. Set to 1 when done. CTNGA(I,1) = phrase number CTNGA(I,2) = action done
CTTIME	I	Time constraint on a message
CTVAR	I	Index of common variable subject to time constraint
CTSAID	IA	Phrase to be spoken, or just spoken
CTINDEX	I	Index of model controller phrase
CTEND	L	True if run is over in demo modes
CTOLDTIME	I	Time phrase was inserted in CTRLR common
CTOLDGPP	I	Old glidepath position
CTIT	I	Message for BXFED for FEED,SAYIT,HOLD
CTOCRT	I	Old turn message
CTSTOP	I	Time to stop turn for no gyro
CTRCL	L	T if waveoff put in CTRLR common (equiv to CTMISS)
CTCALL	I	Votrax phrase number for present call sign
CTBUTTON	I	Votrax phrase number for present button
CTHTBL	IA	Table of correct crabbed headings for turner (0) Final turn heading (1) Second turn toward final (2) First turn toward final
CTOLDOL	R	Previous value of target centerline overlap parameter
CTMSTATE	I	Current state number of turner message automation (0) No prior assigned heading (1) A heading has been assigned but not given to the pilot (2) "Correcting" to be given (only once) (3) Heading and correcting given

CTRLR.CO, (Continued)

Common Variable	Type	Description
CTTSTATE	I	Current state number of turner-disabling automation (1) If turner is free to issue a turn (2) If model controller has suspended further turn advisories for 1/2 mile subsequent to giving a correction once the track is acquired
CTENBRNG	R	Range at which turner will be re-enable (ft)
CTTMODE	I	Current state number of hdg-selection-mode automation (1) If turn to final has not been given (2) If turn to final has been given
CTISIGNX	I	=-1 if left traffic approach; else -1
CTPAST	I	Past time of last picky response
CTGYRO	I	1 – say heading, 2 – say no gyro
CTAPOP	L	True if APGP given
CTCALEXP	I	(1) If ISAY says to call EXPERT (2) If DONE says to call EXPERT
CTOLDCRP	I	Votrax phrase number for last course message output
CTGYTURN	L	True if no gyro turn is in progress

DEV1.CO, CPU 1 Integer Channel Numbers.

Common Variable	F5 Channel	Use	Open
NCSYL	1	Syllabus file	Always
NCSR	2	Student records	Newte
NCPHZ	3	Problem files, all phases also remedial training file	During phase
NCSCR	4	Scratch file-status info	Always
NCLPT	12	Printer	Always
NCVRO	5	Votrax	Always
NCFRZ	13	Fraz file	Always
NCOL	6	Not used	
NCAO	10	Instructor CRT	Always
NCAIN	11	Instructor keyboard	Always
NCRPLY	15	Replay file, radar data	Phase 3
NCRPAT	16	Replay file, activity data	Phase 3 P-run
NCDV	17	Digitized voice file (student)	Phase 3 P-run Phase 1
NCCDV	14	Canned digitized file	Always
NCIDV	20	Index into student digitized voice file	Phase 1
NCBUG	7	Bug file	Always
NCDPI	8	IPB input	Always
NCDPO	9	IPB output	Always
NCERR	18	Student error	P-run
NCERX	19	Error output texts	Phase 2,replay
NCSR1	21	Student performance short summary blocks	Always
NCPV19	22	Student performance PV19	Always
NCSUM	23	Student performance P-3 summary blocks	Always

DEV1.CO, CPU 1 Integer Channel Numbers (continued)

Common Variable	F5 Channel	Use	Open
[illegible]REM	24	Remedial training	TZEC
[illegible]	25	Student performance P-3 blocks	Always
NCPH	30	Text file of SUS	Phrases
NCTMP	31	Channel for temporary use	
NCSR2	32	Student performance long records	Always
NCSTFE	33	Student feedback file	After run, during replay
NCERIN	34	Error index file	Always

DEV2.CO, CPU 2 Integer Channel Numbers.

Common Variable	F5 Channel	Use	Open
NCTXT	1	Text files, all phases	During phase
NCVF	2	Voice data file	After sign on
NCTAR	3	Voice data training arrays	Phase 1
NCAO	10	Student CRT	Always
NCAIN	11	Student keyboard	Always
NCDPI	8	IPB input	Always
NCDPO	9	IPB output	Always
NCBUG	7	Bug file	Always
NCSPK	5	Speech constants	Phase 1
NCOL	6	Overlay file	Always
NCTMP	13	Channel for temp use	

EMERGE.CO, CPU 1 Emergency Data.

Common Variable	Type	Description
EMGYFL	L	True if gyro failure is to occur
EMGYR	R	Range at which gyro failure occurs
EMICFL	L	Not used
EMICR	R	Not used

ENVIRON.CO, CPU 1 Environment Data.

Common Variable	Type	Description
ENCAT	L	Not used
ENSCAT	I	Starting Posn, multipossibility
ENGCAT	I	PTWEEL, Multipossibility
ENVCAT	I	Not used
ENDCAT	I	Not used
ENRH	R	Runway heading (radians magnetic)
ENWHT	R	Mean wind direction (radians magnetic)
ENMWS	R	Mean nogust windspeed along wind hdg (ft/sec)
ENMGS	R	Mean gusting windspeed along wind hdg (ft/sec)
ENMAGS	R	Mean antigust windspeed along wind hdg (ft/sec)
ENMGD	R	Mean gust duration (sec)
ENGOCC	R	Fraction of time gusts occur (.LE.0.5)
ENWVP	R	Wind variability parameter (dimensionless)
ENWSCT	R	Windspeed correlation time (sec)
ENWHR	R	Mean wind direction relative to z-axis (rad)
ENCOS	R	Cosine (ENWHR)
ENSIN	R	Sine (ENWHR)
ENXH	R	Mag hdg positive-x-axis points (rad mag)
ENWMX	R	X-component of mean wind velocity (ft/sec)
ENWMZ	R	Z-component of mean wind velocity (ft/sec)
ENALPHA	R	Windspeed correlation time parameter
ENBETA	R	Windspeed correlation time parameter
ENK3	R	Windspeed autocorrelation parameter
EN2K3	R	Windspeed autocorrelation parameter (=2*ENK3)
ENK4	R	Windspeed autocorrelation parameter
EN2K4	R	Windspeed autocorrelation parameter (=2*ENK4)
ENN1	I	Wind state selection parameter (dimensionless)
ENN2	I	Wind state selection parameter (dimensionless)
ENN3	I	Wind state selection parameter (dimensionless)
ENSEED	I	Seed for wind module random number generator

ENVIRON.CO, CPU 1 Environment Data (Continued)

Common Variable	Type	Description
ENW1	R	Windspeed autocorrelation variable
ENW2	R	Windspeed autocorrelation variable
ENWST	I	Wind state (nogust=1,gust=2,antigust=3)
ENWX	R	X-component of current wind velocity (ft/sec)
EMWZ	R	Z-component of current wind velocity (ft/sec)
ENWSP	I	Current windspeed (kts, nearest non-neg integer)
ENWHDG	I	Current wind heading (deg, nearest integer)
ENSUSP	L	If true ENWSP and ENWHDG freeze
ENCEIL	R	Altitude at current base of overcast (ft)
ENSW	RA	Array containing ENMWS,ENMGS,ENMAGS values

ERR.CO, This block is used by PERRCHK to hold indices into the error explanation file.

Common Variable	Type	Description
ERINDEX	IA	Index from bit-word of PMV to record error of its explanation in ERXFI.
ERRPTR	I	Pointer into ERRFI where next error index should go.

EVNT2.CO, CPU 2 Event Numbers.

Parameter Name	Value	Description
EVPHZ	1	Wakeup phase executive
EVVIN	2	Signal end of student voice input
EVKEY	5	Signal student keyboard input
EVVRPD	3	Signals end of VRP load
EVVST	10	Signal start of voice input
EVKYST	12	Signal end of special processing to student keyboard
EVLVL	13	Level start/stop from VDC
EVIPB	14	IPB-TALKOUT communications

EVNTS.CO, CPU 1 Event Numbers.

Parameter Name	Value	Description
EVPHZ	1	Wakeup phase executive (1,2, or 3)
EVVIN	2	Signal end of student voice input
EVVRO	3	Signal end of Votrax output
EVPNL	4	Signal input from student panel
EVKEY	5	Signal keyboard input (student)
EVZEC	6	Wakeup training executive
EVSPT	7	SPOUT finished
EVRDY	8	Model controller update done
EVSYN	9	End of message syntax
EVVST	10	Signal start of voice input
EVPLT	11	Pilot has output to GLIB
EVKYST	12	Signal end of special processing to instructor kbrd
EVLVL	13	Level1 start/stop from VDC
EVPRC	14	Processing complete from KTEACH, KSTUD to KPROC
EVIPB	15	IPB — taskout communication
EVSPN	16	Signals end of digital voice input from student
EVTXT	17	Signals completion of page display in PLATEXT

FGROUND.CO, Interground Communications Area.

The offsets in this common area are known in the background through the parameters in BGROUND.CO.

Common Variable	Type	Description
FGMSG	I	Message code sent across IPB (IDPICUP, IDSERVO, IDPKSRV, IDIMAGES, or IDDIE) (offset BKMSG) when -1, message has been processed.
FGPCMSG-FGPCHDSP	I	Variables for transfer to PCP.CO (offset BKPICUP)
FGSVMSG-FGSVY	I	Variables for transfer to SRV.CO (offset BKSERVO)
FGIM1, FGIM2	I	Input arguments for IMAGES (offset BKIMAGES)
FGIPB	I	IPBOUT2 message ID, -1 when message processed (offset BKIPB)
FGTMSG	I	IPBOUT2 message
FGSRV	I	Indicator of a message for radar from SERVO (offset BKSRV)
FGSRV1	I	First word of SERVO message (RDSVAZ)
FGSRV2	I	Second word of SERVO message (RDSVEL)
FGSPCL	I	Sync message or stop command (offset BKSPCL)
FGSTRING	I	Megatek string (offset BKSTRING)

Parameter Name	Value	Description
FGNWDS	50	Size of common area

FIL1.CO, CPU 1 File Names.

Common Variable	Type	Description
FNSYL	IA	Syllabus file name
FNDR	IA	Student directory file name
FNTXT	IA	Text file name (to CPU 1)
FNPHZ	IA	Phase problem file names
FNDV	IA	Digitized speech
FNRPLY	IA	Radar replay
RNRPAT	IA	Activity file
FNERR	IA	Student error
FNERX	IA	Error output texts
FNSR1	IA	Student record file #1 name
FNSUM	IA	Student record summary file name
FNP3	IA	Student record phase 3 block file name
FNPV19	IA	Student record PV19 file name
FNFORM1	IA	Formatted output of form 1
FNPDV	IA	P-run digitized voice file
FNPRPL	IA	P-run display replay file
FNPRPAT	IA	P-run activity replay file
FNPERR	IA	P-run error file
FNIFB	IA	Type 1 instructor feedback file name
FNDFALT	IA	Default directory for CPU 1
FNPERR	IA	P-run error file

FIL1.CO, CPU 1 File Names (Continued)

Parameter Name	Value	Description
FNRBLK	25	Number of contiguous blocks in RPLDSP
FNSBLK	1440	Number of contiguous blocks in RPLSPH
FNVRPBLK	26	Number of contiguous blocks in VRP.VO
FNIFPBLK	221	Number of contiguous blocks in IFP.VO
FNLNSUM	64	Record length of summary file
FNLNSCRATCH	48	Record length of scratch file
FNLNSR1	48	Record length of SR1
FNLIDVFILE	4	Record length of RP index file
FNLACT	16	Record length of activity file
FNLERR	8	Record length of error file
FNLDSP	16	Record length of display replay file
FNLP3	240	Record length of P3
FNLERX	86	Record length of error index file
FNSCRATCH	"SCRATCH"	Scratch file name
FNIDFILE	"IDVFILE"	Replay RP index file
FNPIDFILE	"PIDVFILE"	P-run RP index file
FNCIDFILE	"CIDVFILE"	Student's canned RP index file
FNSCANFILE	"SCANFILE"	Student's canned RP file
FNERIN	"ERBLK"	Error description file

FIL2.CO, CPU 2 File Names.

Common Variable	Type	Description
FNIFP	IA	IFP collection file name
FNVRP	IA	VRP collection file name
FNSPK	IA	Trainee independent VDC/SUS
FNTXT	IA	Text file name
FNDR	IA	Student subdirectory
FNDEFALT	IA	Default directory for CPU 2
FNAME	IA	Student's first name

Parameter Name	Value	Description
FNDIR	4	Size of student subdirectory
FNSIZ	12	Size of name array

FZ1.CO, CPU 1 Phase 1 Information.

Common Variable	Type	Description
FZPTR	I	Phase 1 file position pointer
FZINP	I	"A" input format, content of current phase 1 record
FZSKP	I	Number of records to be skipped upon timeout of wait condition
FZFRZ	I	Freeze key message number
FZSUB	IA	Stores five abnormal return skip record values. These skips are in reference to the normal return point. Array allows five levels of nesting.
FZFLG	IA	Holds flag values. Array provides 10 flags to be manipulated by the phase 1 task file.
FZSRV	L	If .true. servo has been activated, if .false. servo is frozen
FZNST	I	Level of instruction file subroutine nesting(1-5)

Parameter Name	Value	Description
FZTOUT	-9999	Timeout message used in P1WAI,P1AZLR

FZEC.CO Foreground Executive Common.

Common Variable	Type	Description
FZPHASE	I	Set to 1 for run in progress, else 2
FZMEGLOCK	L	True when graphics routines are in use

GZEC.CO, CPU 1 GCA-CTS Executive Variables.

Common Variable	Type	Description
GZNR	I	Number of repeats of this problem
GZFRZ	L	Error termination, phase 2
GZTRY	I	Number of tries, this problem, phase 2
GZPHZ	I	Phase
GZPAS	L	Used to inform phase 3 that criteria for advancement have been met
GZSOFL	L	Where true, IPBIN1 sends signoff request to TZEC
GZCHAL	L	Challenge phase 3 (skipping phase 2)
GZSEED	I	Seed for next pseudo-random number
GZRPL	I	Set to type of replay if mandatory replay request was made
GZSKY	I	Digitized speech interrupt service IXMT message key
GZPILL	LA	Remedial training exercises proposed by select.
GZPRUN	L	True when P run is in progress
GZOR	I	Type of override
		-1 – No override
		0 – Repeat old task immediately
		1 – Repeat old task after this problem
		2 – Proceed to next sequential task
GZTASK	IA	Problem to repeat
GZCUR	I	Current task
GZALZN	IA	Proposed alignment problem
GZTIM	I	Counter for student time outs
GZDONE	L	True if between tasks, if false start in middle (P3CHSV) of task (FNPHZ)
GZSUM	L	If true, calculate a phase 3 summary block
GZOPHZ	I	Phase of proposed override task
GZRUN	I	Type of task currently running
		0 – Normal syllabus task
		1 – Override task
		2 – Remedial task

GZEC.CO, CPU 1 GCA-CTS Executive Variables (Continued)

Common Variable	Type	Description
GZMNR	I	The minimum number of repeats to pass present task
GZREM	L	If true, remediation was specified
GZSC19	L	If true, do alignment test
GZPTRY	I	The number of passing approaches (phase 2 only)
GZCSYL	IA	Channel save for position in syllabus file, differs fron TZCSYL in that it points to the position to begin reading if the student does pass
GZADAPTWF	L	True if wind fluctuation to be adapted
GZADAPTPT	L	True if pilot is to be adapted
GZADAPTAC	L	True if air craft type is to be adapted
GZRPT	L	True if this phase 3 problem is to be repeated
GZGO	I	Used by SRMON to know a/c is flying and by TASKOUT to know to start SUS 0 — SUS should not be started 1 — while a/c is on the display 2 — after the run
GZTOT	I	Total number of minutes student has spent on system
GZBRKTIME	I	Number of minutes between "break" messages. Here so it is easy to change per NATTC request
GZNOD3FREE	L	.True. when EXPERT is not in need of node 2 .False. when EXPERT has node 2 checked by IKBRD and TASKOUT before starting KPROC.

Parameter Name	Value	Description
GZPZ1	1	Phase 1 value of GZPH2
GZPZ2	2	Phase 2 value of GZPH2
GZPZ3	3	Phase 3 value of GZPH2
GZPRN	4	P-run value of GZPH2
GZRPLY	5	Replay value of GZPH2
GZDMO	6	Demo value of GZPH2

GZEC.CO, CPU 1 GCA-CTS Executive Variables (Continued)

Parameter Name	Value	Description
GZNOOR	1	No override value of GZOR
GZREADY	0	Override task ready value of GZOR
GZWAITING	1	Override task waiting value of GZOR
GZCONT	2	Continue to next task value of GZOR
GZNORMAL	0	Normal syllabus task value of GZRUN
GZREMEDIAL	1	Remedial task value of GZRUN
GZOVERRIDE	2	Override task value of GZRUN
GZAWAIT	15	Wait for alignment test
GZTWAIT	180	180 sec wait on text file reads
GZWAIT10	10	10 wait for normal timeouts
GZQWAIT	60	Wait for direct questions
GZRWAIT	120	Wait after an approach
GZINSTWAIT	60	Wait for instructor response
GZSTARTCL	1	Start the clock, zero timers
GZRESTARTCL	2	Restart the clock
GZSTOPCL	3	Stop the clock
GZNOSUS	0	SUS should not be started value of GZGO
GZVISIBLE	1	Run has started value of GZGO
GZINVISIBLE	2	A/C is no longer on display value of GZGO

IDEXEC.CO, CPU 1 Parameters for Dummy Argument 'Next'.

Parameter Name	Value	Description
OK	-1	No change
IDEMO	1	Next task is DEMO
IPZDEMO	2	Next task is PZDEMO
ITZEC	3	Next task is TZEC
IPHAZ23	4	Next task is PHAZ23
IPZ23	5	Next task is PZ23
IP2RUN	6	Next task is P2RUN
IP3RUN	7	Next task is P3RUN
IPZ3B	8	Next task is PZ3B
IPHAZ1	9	Next task is PHAZ1
ISGNOFF	10	Next task is SGNOFF

IDPRI1.CO, ID's, Priorities and Stack Sizes of CPU 1 Tasks.

Tasks: Add More ID's at end please, not in alphabetical order.
ID # in octal in comment

Parameter Name	Value	Description of IDs
IDKPROC	1	1: KPROC
IDSUS	2	2: SUS
IDVSPRES	3	3: VSPRES
IDRCVIN	4	4: RCVIN
IDRADAR	5	5: RADAR
IDPHAZ1	6	6: PHAZ1
IDAWAKE	7	7: WAKEUPS
IDSINON	8	10: SINON
IDLEVEL1	9	11: LEVEL1
IDRDCHG	10	12: NOT A TASK
IDKILL	11	13: KILL TASKS
IDOVERRIDE	12	14: OVERRIDE
IDAPEX	16	20: APEX
IDAPRAX	16	20: APRAX
IDAPREX	16	20: APREX
IDDEMO	17	21: DEMO
IDIPBIN1	18	22: IPBIN1
IDPANEL	20	24: PANEL
IDTASKOUT	21	25: TASKOUT
IDIKBRD	22	26: IKBRD
IDKSTUD	23	27: KSTUD
IDKTEACH	24	30: KTEACH
IDINITRT	25	31: INITRT
IDFOR1	26	32: FOR1
IDNEWTE	27	33: NEWTE
IDKREPLA	28	34: KREPLA
IDPRNTIT	29	35: PRNTIT
IDMODIFY	31	37: MODIFY
IDTZEC	33	41: TZEC

IDPRI1.CO, ID's, Priorities and Stack Sizes of CPU 1 Tasks (Continued)

Parameter Name	Value	Description of IDs
IDATRPLY	34	42: ATRPLY
IDRDRPLY	35	43: RDRPLY
IDSPOUT	36	44: SPOUT
IDREPLAY	37	45: REPLAY
IDTIMEOUT	38	46: TIMEOUT
IDZTIM	39	47: ZTIM
IDPHAZ23	40	50: PHAZ23
IDP2RUN	41	51: P2RUN
IDP3RUN	42	52: P3RUN
IDDONE	45	55: DONE
IDISAY	46	56: ISAY
IDEX1PERT	47	57: EX1PERT
IDWRFRAZ	48	60: WRFRAZ
IDPZ23	49	61: PZ23
IDTIMER	50	62: TIMER
IDP1AZLR	51	63: P1AZLR
IDSPBUF	52	64: SPBUF
IDSPDMP	53	65: SPDMP
IDRESPOND	54	66: RESPOND
IDSRMON	55	67: SRMON
IDPZDEMO	56	70: PZDEMO
IDPZ3B	57	71: PZ3B
IDRTZEC	59	73: RTZEC
IDRPZ23	60	74: RPZ23
IDCKIN	61	75: CKIN
IDCKCRP	62	76: CKCRP
IDCKGPP	63	77: CKGPP
IDTGT50	64	100: TGT50
IDFEED	65	101: FEED
IDSHFSTOP	66	102: SHFSTOP
IDSAYIT	67	103: SAYIT

IDPRI1.CO, ID's, Priorities and Stack Sizes of CPU 1 Tasks (Continued)

Parameter Name	Value	Description of IDs
IDAPGP	68	104: APGP
IDENDFEED	69	105: ENDFEED
IDACDMP	70	106: ACDMP (to change ID, change ACTOUT)
IDRDACT	71	107: RDACT
ID3NODE	72	110: ACTUAL ID of node 3 routines (KPROC, DONE)
IDSR1FIN	73	111: SR1FIN
IDSUSWRITE	74	112: SUSWRITE
IDRZEC	75	113: RZEC

Parameter Name	Value	Description of Priorities
IRACDMP	60	ACDMP (to change, change ACTOUT)
IRAPGP	50	APGP
IRATRPLY	50	ATRPLY
IRCYCLIC	40	APEX,APREX,APRAX
IRCKCRP	50	CKCRP
IRCKGPP	50	CKGPP
IRCKIN	50	CKIN
IRDEMO	50	DEMO
IRDONE	45	DONE
IRDPHAZ1	50	DPHAZ1
IRENDFEED	50	ENDFEED
IREX1PERT	50	EXPERT
IRFEED	50	FEED
IRIKBRD	50	IKBRD
IRINITRT	50	INITRT
IRIPBIN1	50	IPBIN1
IRISAY	50	ISAY
IRKPROC	50	KPROC
IRKREPLA	50	KREPLA
IRKSTUD	50	KSTUD
IRKTEACH	50	KTEACH

IDPRI1.CO, ID's, Priorities and Stack Sizes of CPU 1 Tasks (Continued)

Parameter Name	Value	Description of Priorities
IRLEVEL1	50	LEVEL1
IRMODIFY	50	MODIFY
IRNEWTE	50	NEWTE
IR3NODE	50	NODE 3 ROUTINES
IRPANEL	50	PANEL
IRPHAZ1	50	PHAZ1
IRPHAZ23	50	PHAZ23
IRPRNTIT	50	PRNTIT
IRPZDEMO	50	PZDEMO
IRPZ23	50	PZ23
IRPZ3B	50	PZ3B
IRP1AZLR	50	P1AZLR
IRP2RUN	50	P2RUN
IRP3RUN	50	P3RUN
IRRADAR	50	RADAR
IRRCVIN	50	RCVIN
IRRDACT	50	RDACT
IRRDBUF	50	RDBUF
IRRDRPLY	50	RDRPLY
IRREPLAY	50	REPLAY
IRRESPOND	50	RESPOND
IRRPZ23	50	RPZ23
IRRTZEC	50	RTZEC
IRRZEC	50	RZEC
IRSAYIT	50	SAYIT
IRSHFSTOP	50	SHFSTOP
IRSINON	50	SINON
IRSPBUF	50	SPBUF
IRSPDMP	50	SPDMP
IRSPOUT	50	SPOUT
IRSRMON	50	SRMON
IRSR1FIN	50	SR1FIN

IDPRI1.CO, ID's, Priorities and Stack Sizes of CPU 1 Tasks (Continued)

Parameter Name	Value	Description of Priorities
IRFOR1	50	FOR1
IRSUS	50	SUS
IRSUSWRITE	50	SUSWRITE
IRTASKOUT	50	TASKOUT
IRTGT50	50	TGT50
IRTIMEOUT	50	TIMEOUT
IRTIMER	20	TIMER
IRTZEC	50	TZEC
IRVSPRES	50	VSPRES
IRWRFRAZ	50	WRFRAZ
IRZTIM	50	ZTIM

Parameter Name	Value	Description of Stack Partitions
RTACDMP	60	ACDMP (to change, change ACTOUT)
RTAPEX	0	APEX
RTAPGP	0	APGP
RTAPRAX	0	APRAX
RTAPREX	0	APREX
RTCKIN	0	CKIN
RTCKCRP	0	CKCRP
RTCKGPP	250	CKGPP
RTCKIN	250	CKIN
RTDONE	0	DONE
RTENDFEED	0	ENDFEED
RTEX1PERT	0	EX1PERT
[illegible]	0	FEED
[illegible]	250	FOR1
[illegible]	250	IKBRD
[illegible]	250	IPBIN1
[illegible]	150	ISAY

IDPRI1.CO, ID's, Priorities and Stack Sizes of CPU 1 Tasks (Continued)

Parameter Name	Value	Description of Stack Partitions
RTKPROC	0	KPROC
RTLEVEL1	150	LEVEL1
RTP1AZLR	0	P1AZLR
RT3NODE	0	NODE 3 ROUTINES
RTPANEL	300	PANEL
RTPZ23	60	PZ23
RTRDACT	0	RDACT
RTRDBUF	60	RDBUF
RTRPZ23	0	RPZ23
RTRTZEC	0	RTZEC
RTRZEC	0	RZEC
RTSAYIT	0	SAYIT
RTSHFSTOP	0	SHFSTOP
RTSPBUF	60	SPBUF
RTSPDMP	60	SPDMP
RTSUSWRITE	0	SUSWRITE
RTFOR1	0	FOR1
RTTASKOUT	300	TASKOUT
RTTGT50	250	TGT50
RTTIMEOUT	60	TIMEOUT
RTTZEC	60	TZEC
RTZTIM	60	ZTIM
RTTZEC	0	TZEC
RTPZ23	0	PZ23
RTPHAZ23	0	PHAZ23
RTP3RUN	0	P3RUN
RTP2RUN	0	P2RUN
RTTIMER	60	TIMER
RTDEMO	0	DEMO
RTPZDEMO	0	PZDEMO
RTSRMON	0	SRMON
RTPZ3B	0	PZ3B
RTSR1FIN	0	SR1FIN

IDPRI2.CO, ID's With Priorities and Stack Sizes of CPU 2 Tasks.

Parameter Name	Value	Description of IDs
IDMEGSTR	1	1: Megatek strings (not a task)
IDIMAGES	2	2: IMAGES (not a task)
IDPICUP	3	3: PICUP (not a task)
IDPKSRV	4	4: PICUP/SERVO (not a task)
IDSERVO	5	5: SERVO
IDTEXT	6	6: Text file (PLATEXT)
IDSTIFLE	7	7: STIFLE
IDTIME	8	10: Time sync (not a task)
IDSPEECH	9	11: SPEECH
IDLEVEL	10	12: LEVEL
IDHEARSAY	11	13: HEARSAY
IDDIE	13	15: Stop (not a task)
IDSKPRO	14	16: SKPRO
IDPRESENT	15	17: PRESENT
IDMENU	16	20: MENU (not a task)
IDSTUDSTATS	17	21: STUDSTATS
IDKILLTSK	18	22: Kill tasks (not a task)
IDSKBRD	19	23: SKBRD
IDHELLO	20	24: HELLO
IDINIT2RT	21	25: INIT2RT
IDCRT	22	26: Text string (not a task)
IDSTOVER	23	27: STOVER
IDLOOKOUT	28	34: LOOKOUT
IDSAID	29	35: SAID
IDIPBIN2	31	37: IPBIN2
IDTALKOUT	33	41: TALKOUT
IDVAL2WT	34	42: VAL2WT
IDVRPLD	35	43: VRPLD
IDVSRRC	36	44: VSRRC

IDPRI2.CO, ID's With Priorities and Stack Sizes of CPU 2 Tasks (Continued)

Parameter Name	Value	Description of IDs
IDCKCMN	37	45: CKCMN
IDLOOKFORWARD	38	46: LOKFORWARD
IDNODE3	39	47: Actual task ID of node 3 routines: STIFLE, SKPRO, HELLO, PLATEXT, INIT2RT STUDSTATS, STOVER
IDSERVUP	40	50: SERVUP

Parameter Name	Value	Description of Priorities
IRCKCMN	50	CKCMN
IRHEARSAY	50	HEARSAY
IRHELLO	50	HELLO
IRINIT2RT	50	INIT2RT
IRIPBIN2	50	IPBIN2
IRLEVEL	50	LEVEL
IRLOOKFORWARD	50	LOKFORWARD
IRLOOKOUT	50	LOOKOUT
IRPRESENT	50	PRESENT
IRSAID	40	SAID
IRSERVUP	50	SERVUP
IRSKBRD	50	SKBRD
IRSKPRO	50	SKPRO
IRSPEECH	50	SPEECH
IRSTIFLE	50	STIFLE
IRSTOVER	50	STOVER
IRSTUDSTATS	50	STUDSTATS
IRTALKOUT	50	TALKOUT
IRTEXT	50	PLATEXT
IRVAL2WT	40	VAL2WT
IRVRPLD	50	VRPLD
IRVSRRC	50	VSRRC
IRIMAGES	50	IMAGES
IRPICUP	50	PICUP

IDPRI2.CO, ID's With Priorities and Stack Sizes of CPU 2 Tasks (Continued)

Parameter Name	Value	Description of Stock Part Items
RTCKCMN	150	CKCMN 100 is ok
RTHEARSAY	100	HEARSAY
RTHELLO	0	HELLO
RTINIT2RT	0	INIT2RT
RTIPBIN2	250	IPBIN2
RTLEVEL	55	LEVEL
RTLOOKFORWARD	300	LOKFORWARD
RTLOOKOUT	300	LOOKOUT
RTSAID	150	SAID
RTSERVUP	0	SERVUP
RTSTIFLE	100	STIFLE
RTSTOVER	0	STOVER
RTSTUDSTATS	0	STUDSTATS
RTTALKOUT	300	TALKOUT
RTTEXT	0	PLATEXT
RTVRPLD	150	VRPLD default stack
RTVSRRC	150	VSRRC

IPB1STF.CO, Common for IPB on CPU 1

Common Variable	Type	Description
IPNUM	I	Number of arguments
IPARGNO	IA	Number of legal arguments indexed by task ID. Circular queue.
IPARGS	IA	Arguments to send to the tasks
IPFILL	I	Fill pointer, IPARGS
IPTID	IA	Task ID holder

Parameter Name	Value	Description
IPNBLK	10	Size of each block in IPARGS
IPNARG	10*IPNBBLK-1	Upper bound of IPARGS
IPNBL	(IPNARG+1)1 IPNBLK-1	Number of blocks in IPARGS-1
IPALIM	25	High limit for number of arguments

IPBSTF.CO, IPB Common for CPU 2.

Common Variable	Type	Description
IPNUM IPARGNO	IA	Number of arguments legal for each task, indexed by ID
IPARGS	IA	Arguments to be sent to each task
IPFILL	I	Fill pointer, IPARGS
IPSTRING	IA	String holder
IPTID	IA	Task ID holder

Parameter Name	Value	Description
IPNBLK	10	Size of each block in IPARGS
IPNARG	10*IPNBLK-1	Upper bound of IPARGS
IPNBL	(IPNARG+1)/ IPNBLK-1	Number of blocks in IPARGS-1
IPALIM	15	Max task ID
IPARGNOIMAGES	2	CPU 1 IPARGNO (IDIMAGES)
IPARGNOPICUP	7	CPU 1 IPARGNO (IDPICUP)
IPARGNOSERVO	3	CPU 1 IPARGNO (IDSERVO)
IPARGNOPKSRV	10	CPU 1 IPARGNO (IDPKSRV)

KBRD.CO, CPU 1 Keyboard Variables.

Common Variable	Type	Description
KBINST	L	T if instructor functions active on T/E kbrd
KBFRZ	L	T if wait or abort was pressed. Changes menu.
KBNUM	L	T if validation not to take place
KBBYE	L	T if bye was pressed on student side
KBITBL	L	T if address table already set up for dispatch on instructor side
KBINUM	L	T if validation not to take place on instructor side
KBSTBL	L	T if address table already set up for dispatch on student side
KBTALK	L	T if init voice test was pressed. Changes menu
KBLST	I	Last key pressed on student side
KBSLO	I	Lowest legal switch on address table, student
KBSHI	I	Highest legal switch on address table, student
KBSTABLE	IA	Address switches table for dispatch, student
KBST	IA	Student keys array
KBIN	IA	Instructor keys array
KBSBIT	I	Bit for correct menu for student
KBIBIT	I	Bit for correct menu for instructor
KBILO	I	Lowest legal switch on address table, instructor
KBIHI	I	Lowest legal switch on address table, student
KBITABLE	IA	Address switches tables for dispatch, instructor
KBIMENU	IA	Text for instructor menu
KBSMENU	IA	Text for student menu
KBVSTRT	L	Start voice test if T
KBVSTOP	L	Stop voice test if T
KBALIGN	L	T if ALIGN was pressed
KBSHFST	L	T if SHIFT STOP was pressed
KBWAIT	L	T if WAIT was pressed
KBCONT	L	T if CONT was pressed
KBABORT	L	T if ABORT was pressed

KBRD.CO, CPU 1 Kevboard Variables (Continued)

Common Variable	Type	Description
KBTYPE	I	0 – CTRL/C enabled, 1 – CTRL/C disabled
KBSMSG	IA	Messages for SKPRO indexed by student key
KBIMSG	IA	Messages for SKPRO indexed by instr key
IBOR	I	For instructor overrides
KBMODIFY	L	Replay modify request received
KBPRNTIT	L	Printout requests received
KBLNEWRT	L	New R/T request received
KBLREPLA	L	P-run replay request received
KBLNEWTE	L	New trainee request received
KBLHELLO	L	SIGNON is complete
KBLOVERRIDE	L	Override request pending
KBLHERE	L	Instructor is responding to questions
KBLFOR1	L	True if stats request received
KBMENU	I	113:MENU
KBHELP	I	114:HELP
KBINVT	I	115:INIT VOICE TEST
KBSTVT	I	116:STOP VOICE TEST
KBALGN	I	117:ALIGN
KBNEXT	I	118:NEXT
KBYES	I	119:YES
KBNO	I	120:NO
KBSTATS	I	121:STATS
KBHELLO	I	122:HELLO
KBYE	I	123:BYE
KBCTRLC	I	53:CTRLC
KBEXIT	I	46:EXIT T/E KBRD
KBFREEZE	I	55:WAIT
KBPROCEED	I	52:CONT
KBORT	I	49:ABORT
KBOVER	I	48:OVERRIDE
KBNEWRT	I	56:INIT NEW R/T
KBSR	I	50:SR

KBRD.CO, CPU 1 Keyboard Variables (Continued)

Common Variable	Type	Description
KBREPLA	I	57:REPLA
KBNEWTE	I	114:NEW T/E
KBMOD	I	54:MOD
KBPRINT	I	122:PRINT STATS
KBSTOP	I	107:SHIFT STOP
KBINTE	I	51:INIT T/E KYBD

Parameter Name	Value	Description
KBNIK	19	Number of legal instructor keys
KBNSK	11	Number of legal student keys
KBTIMEOUT	-1	Code for timeout
KBNSP	25	Number of keys
KBPBYE	11	Index into KBSMSG for bye message
KBPNRT	14	Index into KBIMSG for INITRT messsage
KBMNVT	8	Menu bit numbers: after INIT VT
KBMNDMO	9	Demo phase prior to SIGNON
KBMNSGN	7	After SIGNON
KBMNWAIT	6	After WAIT
BMNABRT	5	After ABORT
KBMNSTRT	4	After START
KBFROMSKB	0	Keyboard entry from student keyboard
KBFROMIKB	1	Keyboard entry from instructor keyboard
KBFROMOTHER	2	Not a keyboard entry

KBRD2.CO, CPU 2 Keyboard Variables.

Common Variable	Type	Description
KBINST	L	T if instructor functions active on T/E KBRD
KBFRZ	L	T if wait or abort was pressed. Changes menu.
KBLST	I	Last key pressed on student side
KBSBIT	I	Bit for correct menu for student
KBIBIT	I	Bit for correct menu for instructor
KBIMENU	IA	Text for instructor menu
KBSMENU	IA	Text for student menu
KBTYPE	I	C – CTRL C enabled, 1 – CTRL C disabled
KBNUM	L	T if key input is not to be sent to the
KBIN	IA	Potentially valid instructor keyboard entries
KBST	IA	Potentially valid student keyboard entries
		keyboard processor. A wakeup is issued for CPU 2

Parameter Name	Value	Description
KBNIK	19	Number of legal instructor keys
KBNSK	11	Number of legal student keys
KBGZDMO	9	Demo menu bit

KEYS.CO, CPU 1 Indicators of Buttons etc. Depressed at Trainee and Instructor Stations.

Common Variable	Type	Description
KYHOLD	L	When true, interrupt service does not respond to buttons
KYDIA	I	DIA word from student panel
KYDIB	I	DIB word
KYIC3	L	True while ICS button 3 is selected
KYIC5	L	True while ICS button 5 is selected
KYIC7	L	True while ICS button 7 is selected
KYICS	L	True while ICS button SUPER is selected
KY27F	L	True while radio frequency 270.8 is selected
KY31F	L	True while radio frequency 318.8 is selected
KY27M	L	True while radio monitor 270.8 is selected
KY31M	L	True while radio monitor 318.8 is selected
KYREQ	L	True when clearance is requested
KYMIKE	L	True while mike is keyed
KYLVL	I	Level of speech input
KYCLR	L	True while clearance light is on
KYWOL	L	True while waveoff light is on
KYWOB	L	True when waveoff button is depressed
KYSPH	L	True while student is speaking
KYVRO	L	True while $VRO is speaking
KYDOA	I	DOA word
KYDOB	I	DOB word
KYALRM	I	A bit set for everyone who wants alarm on: bit 4-270F; 5-318F; 12-other; 13-w/o
KYFIL	I	PIN fill pointer into KYARY
KYBLD	I	PANEL bleed pointer into KYARY
KYARY	I	Array of panel changes for output to activity file
KYNCC	I	Number of 0.5-sec ticks since last evdnce of "contact"
KYLGA	I	PANLOG DOA storage

KEYS.CO, CPU1 Indicators of Buttons etc. Depressed at Trainee and Instructor Stations (Continued)

Common Variable	Type	Description
KYLGB	I	PANLOG DOB storage
KYMINLVL	I	Minimum KYLVL audible to pilot
KYTALK	L	A logical which tells digitized spch to record

Parameter Name	Value	Description
KY3R	0	DOA — button 3 amber bit
KY3F	1	DOA — 3 flashing
KY5R	2	DOA — 5 amber
KY5F	3	DOA — 5 flashing
KY7R	4	DOA — 7 amber
KY7F	5	DOA — 7 flashing
KYSR	6	DOA — super amber
KYSF	7	DOA — super flashing
KY2A	8	DOA — 270 freq amber
KY2G	9	DOA — 270 freq green
KY3A	10	DOA — 318 freq amber
KY3G	11	DOA — 318 freq green
KYAL	12	DOA — alarm
KY2M	13	DOA — 270 monitor
KY3M	14	DOA — 318 monitor
KYRQ	15	DOA — request
KYCL	0+16	DOA — cleared
KYWR	1+16	DOA — W/O red flashing
KYICR	2+16	DOA — ICS amber
KYICF	3+16	DOA — ICS flashing
		PANIT, PANOUT flags
KYON	1	Enable panel, turn bit on
KYOFF	0	Disable panel, turn bit off
KYSZAR	19	Upper bound in pin-panel circular buffer

MAIL1.CO, CPU 1 XMT/REC Mailboxes.

Common Variable	Type	Description
BXSRMN	I	Not used
BXIPB	I	Not used
BXRPL	I	User clock to RDRPLY IXMT key
BXSPH	I	SPDMP or SPBUF end of digitized replay
BXACT	I	User clock to ATRPLY IXMT key
BXPV	I	Not used
BXRAT	I	Not used
BXRC	I	SPDR interrupt service to SPDMP IXMT key
BXPLY	I	SPDR interrupt service SPBUF IXMT key
BXCYC	I	User clock to APE IXMT key
BXPAN	I	Panel driver to PANEL IXMT key
BXFZ1	I	Phase 1 key; used in P1AZLR, P1WAI, TIMER
BXFED	I	TIMEOUT to FEED routine
BXRZ	I	APE to RNGCAL

MAIL2.CO, CPU 2 REC/XMT Mailboxes.

Common Variable	Type	Description
BXREC	I	Signals voice input reception or timeout message
BXCOG	I	Signals voice recognition made or timeout message
BXTIM	I	Used to sync PICUP in foreground

MENU1.CO, CPU 1 Menus.

Common Variable	Type	Description
MNST	IA	Student menu
MNIN	IA	Instructor Menu
MNINS	IA	Instructor menu for trainee keyboard

AD-A087 190 LOGICON INC SAN DIEGO CA TACTICAL AND TRAINING SYSTE--ETC F/G 17/9
GROUND CONTROLLED APPROACH CONTROLLER TRAINING SYSTEM (GCA-CTS)--ETC(U)
JUN 80 G BARBER, J BOLLENBACHER, D BREWTON N61339-77-C-0162

UNCLASSIFIED NAVTRAEQUIPC-77-C-0162-3 NL

7 OF 8
AD
A087190

MENU2.CO, CPU 2 Menus – A Reflection of Menu1.

Common Variable	Type	Description
MNST	IA	Student menu
MNIN	IA	Instructor menu

MSTRING.CO, CPU2 Megatek String Common.

Common Variable	Type	Description
MSTRING	IA	Array holds string to be written to the Megatek

PARM1.CO, This is the Parameter Common Block for CPU 1.

Parameter Name	Value	Description
NVRP	107	Number of phrases to be recognized
TESTPATTERN	52525K	CPU communication test pattern
PVNM	19	Number of PMS variables

PARM2.CO, This is the Parameter Common Block for CPU 2.

Parameter Name	Value	Description
NVRP	107	Number of phrases to be recognized
TESTPATTERN	52525K	CPU communication test pattern
P2STRINGSIZE	41	Size of arrays for strings

PCP.CO, CPU 2 Aircraft Update Information.

Common Variable	Type	Description
PCTOUCH	I	Label for touchdown reflector on elevation
PCCLEARTOUCH	I	Label for touchdown and clr on azimuth
PCZATAR	I	Intensity of azimuth target
PCZETAR	I	Intensity of elevation target
PCZATRL	I	Intensity of azimuth trail
PCZETRL	I	Intensity of elevation trail
PCX1	I	Megatek coordinates for range, pass 1
PCALTCT	I	Long trail count for azimuth
PCY1	I	Megatek coordinates for top of azimuth target
PCY2	I	Megatek coordinates for bottom of azimuth target
PCY3	I	Megatek coordinates for top of elevation target
PCY4	I	Megatek coordinates for bottom of elevation target
PCMSG	I	MSG code, 1-short trails, 2-long trails
PCATRL	I	Label for azimuth trail
PCETRL	I	Label for elevation trail
PCATAR	I	Label for azimuth target
PCETAR	I	Label for elevation target
PCPAR	IA	Picture array
PCWDIR	I	Wind direction
PCWSPD	I	Wind speed
PCSPW	I	Label for wind
PCNUM	IA	ASCII number set plus blank to set up wind for display
PCSTRNG	I	Label for string of characters to be typed on Megatek
PCAHSH	I	Label for azimuth hashmarks
PCEHSH	I	Label for elevation hashmarks
PCAZANT	I	Label for azimuth cursor
PCELANT	I	Label for elevation cursor
PCAZM	I	Label for azimuth picture

PCP.CO, CPU 2 Aircraft Update Information (Continued)

Common Variable	Type	Description
PCELV	I	Label for elevation picture
PCELDV	I	Label for elevation advisories
PCAZDV	I	Label for azimuth advisories
PCSWP	I	Label for sweep
PCWND	I	Label for wind
PCLNGX	I	Label for long azimuth trail
PCLNGY	I	Label for long elevation trail
PCHG	I	Factor by which display is moved to avoid phosphor burn
PCON	IA	Array of pictures 0 -on, 1 -off, dimension must be equal to PCMAX
PCOLDX	I	Last X1 passed by RADAR
PCY1OLD	I	Last Y1 passed by RADAR
PCY3OLD	I	Last Y3 passed by RADAR
PCCNT	I	Count for long trails
PCASTRT	I	Start of azimuth trail
PCESTRT	I	Start of elevation trail
PCAPTR	I	Pointer into azimuth trail
PCEPTR	I	Pointer into elevation trail
PCSRV	I	Label for servo
PCELTCT	I	Long trail count for elevation

PCP1.CO, Picture Selection Parameters for CPU 1.

Parameter Name	Value	Description
PCMAX	18	Maximum number of pictures
PCAZIM	1	Azimuth display picture #1
PCAZHASH	2	Azimuth hashmarks picture #2
PCELEV	3	Elevation display picture #3
PCELHASH	4	Elevation hashmarks picture #4
PCSWEEP	7	Sweep picture #7
PCAZTARG	8	Azimuth target picture #8
PCAZTRAIL	9	Azimuth trail picture #9
PCELTARG	10	Elevation target picture #10
PCELTRAIL	11	Elevation trail picture #11
PCSTRING	12	ASCII string picture #12
PCWIND	13	Wind picture #13
PCXLONG	14	Azimuth long trail picture #14
PCYLONG	15	Elevation long trail picture #15
PCTDR	16	Centerline reflector on elevation picture #16
PCCLRTDR	17	Touchdown reflector and centerline reflector on azimuth picture #17
PCSERVO	18	Servo, picture #18
PCOFFPIC	0	Turn picture off
PCONPIC	1	Turn picture on
PCRESET	2	Turn off and reinitialize pictures
PCUPDATE	4	Update aircraft
PCSTART	5	Start display processor
PCSTOP	6	Stop display processor
PCTYPE	7	Type string on Megatek
PCFADE	8	Fade trails
PCUPWIND	9	Update wind
PCINIT	10	Initialize display
PCNORMAL	1	Normal PICUP message during run, replay
PCENDOFRUN	2	PICUP message indicating end of run
PCAZCLR	1	Change azimuth servo, and centerline reflector
PCAZTDR	2	Change azimuth servo and touchdown reflector
PCETDR	3	Change elevation servo and touchdown reflector
PCAZFRZ	4	Freeze azimuth servo

PCP1.CO, Picture Selection Parameters for CPU 1 (Continued)

Parameter Name	Value	Description
PCEFRZ	5	Freeze elevation servo
PCMOVE	6	Move hashmarks where desired
PCGRDT1	7	Move hashmarks by where servo is, as in a run
PCSRMON	8	Move hashmarks where SRMON says to

PDIGT.CO. This common file is used primarily to hold strings as parameters for the digitized input routines used in phase 1.

Parameter Name	Value
PDINTRO	'<012><007>Please speak the following phrase:<012><015>'
PDTHANX	'<012> Thank you<15>'
PDRPLAY	'<012>The phrase just spoken will now be replayed..<15>'
PDOK	'<012>Was the recording satisfactory? (Yes or no)<15>'
PDSCOLD	'<012><007><012><012> Hello? Are you there?<015>'
PDOVER	'<012>OK, try again.<15>'

PFSCR.CO, CPU 1 Current PMV Switches.

Common Variable	Type	Description
PFS01	L	True if phase 2 to freeze on error of PV01, or if phase 3 to score FPV01 first.
PFS02	L	True if PV02 score first or freeze on error
PFS03	L	True if PV03 score first or freeze on error
.		
.		
.		
PFS19	L	True if PV19 score first or freeze on error
PFS	LA	Equivalenced to PFS01 – PFS19
PFQUE	IA	Holds the queue of controller messages to be started by put
PFHDG	I	Last assigned heading
PFGPP	I	Last glidepath position given
PFGPT	I	Last glidepath trend given

PLAY.CO, CPU 1 Replay Common.

Common Variable	Type	Description
RPTSP	I	Time speech output to stop, .5 sec ticks

Parameter Name	Value	Description
RPNULL	32000	Used when there are no more errors to be reported
RPMXHD	8	Maximum number of record header types

PLT.CO, Pilot Variables.

Common Variable	Type	Description
PTYP	I	Pilot type code
PTFLT	I	Flight type code (normal,restricted AZ or elev)
PTUZN	I	If restricted flight, right/upper zone number
PTLZN	I	If restricted flight, left/lower zone number
PTAPR	I	Approach type code
PTLOW	R	Range within which low alt alert occurs (ft)
PTRNG	R	Range at which approach terminates (ft)
PTBYE		Not used
PTNGR	L	True if pilot has copied "no gyro apprch" advisory
PTWEEL	I	Used to specify who gives wheels down message: 1 (PTPLTWH): pilot says wheels down 2 (PTMODWH): Model or student controller to say wheels down 3 (PTWHSPK): wheels message spoken
PTGPADSBL	L	True if A/C disabled to produce low-alt alert
PTSEED	I	Seed for plt random number generator
PTREPLY	L	True if pilot ready to make verbal reply
PTRPHRASE	I	Potential-verbal-reply type code
PTRHDG	I	HDG accmpnyng potential-verbal-reply, if any (deg)
PTVPHR	I	Actual-verbal-reply type code
PTVHDG	I	HDG accmpnyng actual-verbal-reply, if any (deg)
PTRESP		Not used
PTPATH		Not used
PTCORS		Not used
PTRANG		Not used
PTNOCOPY	I	Random advisory disregard parameter
PTLCHK	LA	PTLCHK(I) true if LVL check I 0.5-sec ticks from now

PLT.CO, Pilot Variables (Continued)

Common Variable	Type	Description
PT3WBC	LA	PT3WBC(I) true if WBC check I 0.5-sec ticks from now
PTC3WBC	I	Last PTC3WBC advisories were WBC
PTWBC	L	True if Plt decides CTRLR is WBC
PTLAC	L	True if Plt decides CTRLR is LAC
PTMAXNCC	I	If KYNCC GT PTMAXNCC pilot waves himself off
PTASM	R	Plt attempting to maintain this airspeed (ft/sec)
PTA1AS	R	Airspeed tracking error correlation parameter
PTA2AS	R	Airspeed tracking error correlation parameter
PTKAS	R	Airspeed tracking error parameter
PTMAS	R	Airspeed tracking error parameter
PTYDM	R	Plt attempting to maintain this R.O.C. (ft/sec)
PTA1YD	R	R.O.C. tracking error correlation parameter
PTA2YD	R	R.O.C. tracking error correlation parameter
PTKYD	R	R.O.C. tracking error parameter
PTMYD	R	R.O.C. tracking error parameter
PTHDM	R	Plt attempting to maintain this R.O.T. (rad/sec)
PTA1HD	R	R.O.T. tracking error correlation parameter
PTA2HD	R	R.O.T. tracking error correlation parameter
PTKHD	R	R.O.T. tracking error parameter
PTMHD	R	R.O.T. tracking error parameter
PTTR	L	True if plt is turning right, no hdg assigned
PTTL	L	True if plt is turning left, no hdg assigned
PTTRH	L	True if plt is turning right to assigned hdg
PTTLH	L	True if plt is turning left to assigned hdg
PTHMTN	R	Assigned heading for current turn (rad magnetic)
PTHDASS	R	Current assigned R.O.T. all turns (rad/sec magnetic)
PTINCASS	R	Current assigned R.O.T. all turns (rad/half-sec magnetic)
PTTINC	R	Current assigned R.O.T. all turns (rad/half-sec, magnetic, signed)

PLT.CO, Pilot Variables (Continued)

Common Variable	Type	Description
PTASFA	R	Standard final approach airspeed (ft/sec)
PTDES	L	True if plt has copied "begin descent" advisory
PTSYDI	R	Standard initial rate-of-ascent (ft/sec)
PTSVARYDI	R	Variance associated with PTSYDI (ft/sec)**2
PTNOCLR	L	True if correct clearance not yet copied by pilot
PTSPLIT	L	Setting this variable true causes immediate waveoff
PTDELY	RA	Vert disp from G/P implied by adv (as pct of BLPSZE)
PTDLYVAR	RA	Variance associated with PTDELY (ft)**2
PTEYCLK	I	0.5-sec ticks since last G/P adv
PTOEDELY	R	Value of PTDELY at last adv (ft)
PTOEVARDELY	R	Value of PTDLYVAR at last adv (ft)**2
PTYDI	R	Plt est of "ideal R.O.D." (ft/sec)
PTVARYDI	R	Variance associated with PTDYI (ft/sec)**2
PTYDCOR	R	R.O.D. corrctn implied by advisory (ft/sec)
PTMAXYDI	R	Plt beats it if YDI gt this
PIMINYDI	R	Plt beats it if YDI lt this
PTCLO	L	True if plt is climbing now
PTMISS	L	True if MSSD apprch this apprch
PTBEATIT	L	True if plt waved himself off this apprch
PTYMTN	R	Assgnd alt this climb (ft)
PTTRWM	L	True if plt is to turn right upon rchng assgnd alt
PTTLWM	L	True if plt is to turn left upon rchng assngd alt
PTWMAH	R	Assgnd hdg upon rchng assgnd alt (rad magnetic)
PTASCLO	R	Standard climbout airspeed (ft/sec)
PTYDCLO	R	Standard climbout R.O.C. (ft/sec)
PTASPAT	R	Standard pattern airspeed (ft/sec)
PTLODOWN	L	True when aircraft reaches low altitude alert point

PLT.CO, Pilot Variables (Continued)

Parameter Name	Value	Description
PTPLTWH	1	Pilot to give wheels message
PTMODWH	2	Controller to give it
PTWHSPK	3	Somebody said it

PMSSUP.CO. This common block for CPU 1 contains common variables used by PMS support routines.

Common Variable	Type	Description
PSTGT50	L	If T, should execute TGT50
PSEZN	I	Last A/C elevation zone
PSAZN	I	Last A/C azimuth zone
PSMIL	R	Mile recorded at entry into zones 1 or 2
PSCKCRP	L	If T, execute CKCRP
PSCKGPP	L	If T, execute CKGPP

PMVC.CO, CPU 1 PMV Scores.

Mnemonic	Type	Content
PVN	IA	Allowable error scores
PVE	IA	Observed error scores
PV00	I	All things that need to be done on every approach
PV01	I	Handoff composite
PV02	I	Radio check composite
PV03	I	Turn to final composite
PV04	I	Approaching glidepath composite
PV05	IA	Heading advisories composite
PV06	IA	Azimuth position and trend composite
PV07	IA	Glidepath position and trend composite
PV08	IA	Range call composite
PV09	I	Decision height composite
PV10	I	Clearance composite
PV11	I	Landing threshold composite
PV12	I	Handoff, rollout composite
PV13	I	No gyro composite
PV14	IA	No gyro heading corrections
PV15	I	Emergency wave offs
PV16	I	Low altitude alert
PV17	IA	Transmission break
PV18	IA	Transmission rate composite
PV19	I	Radar alignment composite
PVADHG	L	"At decision height" given?
PVACKG	L	Acknowledgement given to pattern?
PVHNG	L	"How..now" given?
PVRCO	L	Radar contact given?
PVRCH	L	Radio check given?
PVGMR	L	"Give me.." request made?
PVMKB	L	Was mike re-keyed?
PVMIK	L	Are we checking keying of mike?
PVTRG50	L	Has 50% of target appeared yet?

PMVC.CO, CPU 1 PMV Scores (Continued)

Mnemonic	Type	Content
PVP1M	L	Is range <= 1 mile?
PVWEEL	L	Has "wheels..down" been given?
PVFPOS	L	Final course position given?
PVOVR	L	"Over" spoken after final course pos.?
PVOLT	L	"Over landing threshold" been given?
PVWNDG	L	Has "wind.." been given?
PVCLRG	L	Clearance given to pilot?
PVWOG	L	Has a wave-off been given?
PVROG	L	Have roll-out instructions been given?
PVBXCG	L	"Button X clear" given yet?
PVHOG	L	Handoff given to pattern controller?
PVHOSC	L	Has CKHO been scheduled?
PVDNA	L	Do-not-acknowledge given?
PVCAN	L	Has tower cancelled clearance?
PVFO	L	Was target far off a cursor at ADH?
PVAGT	I	Time 'approaching glidepath' given in 1/2 secs
PVOTIM	I	Time 'over' given after OLT advisory
PVMSG	I	Last SUS message received (phrase #)
PVCREQ	I	# clearance requests made to tower
PVSTAT	I	Present state of PMS (= 1 or 2)
PVSUB	IA	P** routines called by PSUS
PVNWO	I	Phrase-count of wave-off message
PVEWO	I	A wave-off expected?
		;0: no.
		;1: yes; target not touching at ADH.
		;2: yes; radar contact lost.
PVNEX	I	Bits are used by some P** routines to indicate when they are interested in the next SUS phrase*
PVTOD	L	Target presently off display?
PV2P8	L	Targets w/in 2.8 mi. of each other?
PVNGA	L	No-gyro-approach given?
PVWARN	L	Has 'heading xxx' been given?

PMVC.CO, CPU 1 PMV Scores (Continued)

Mnemonic	Type	Content
PV1HAF	L	'Make 1/2 std rate turns' given?
PVDIS	I	Saz when 'heading xxx' given
PVMILE	I	SAMILE when 'heading xxx' given
PVNCOR	I	# no-gyro heading corrections made
PVTFC	L	Is turn-to-final complete?
PVTW	I	Width of cursor in feet
PVNT	I	# turn-to-final advisories made
PVTRND	L	Glidepath trend message given since last position message?
PVLPM	I	Last glidepath pos. message given
PVGPP	L	Glidepath call given for zone
PVLAAG	L	Low alt. alert given
PVGTN	L	No-gyro turn given
PVLTIME	I	Time OLT given
PVNGT	I	Mile at which normal turn given
PVMISS	L	A/C is on waveoff
PVLGP	I	Last GP zone position given
PVEXE	I	PVNEX routines
PVOTG	L	"On the go" given
PVRATE	I	PV18 pause timer
PV1SB	I	PST1 routines
PVWAV	I	Waveoff routines
PVPMS	L	If true, PMS running for phase 3
PVTURN	I	PTURN routines
PVSP1	I	PSPEC type 1 routines
PVSP2	I	PSPEC type 2 routine
PVSP3	I	PSPEC type 3 routine
PVPCH	I	PSPCH routine
PVDAMSK	I	Mask for bits of interest in DOA word
PVDBMSK	I	Mask for bits of interest in DOB word
PVLOW	L	Low altitude condition holds

PMVC.CO, CPU 1 PMV Scores (Continued)

Mnemonic	Type	Content
PVLP19	I	P19 scoring phase: 0-done, 1-checking servoing procedure, 2-servoing finished

Parameter Name	Value	Description
PVTW6	90	6 mi. display coord. for AZ. TGT width
PVTW3	370	Same as above for 3 miles
PVACURSOR	-863	Note actual cursor display coord.
PVERRATE	10	Acceptable short pause error
PVNONE	0	Initialized for no action
PVMORE	-1	Requires further special action
PVSIZE	266	Present size of PMVC area to init. Begins with *PV00*, ends at *PVLOW*
PVPVS	70	Size of PMVC block occupied by PV00-PV19
PVCMT	BYTE (PVNWO,1)	Phrase expected after "climb and maintain..."
PVCMV	BYTE (PVNWO,2)	Error bit for the above

PMVC.CO, CPU 1 PMV Scores (Continued)

NOTE: All of the following contain either the address of the routine named or 0, if no action is to be taken.

PST1 Routines (PV1SB)

1: P01D

2: P02A

PVWAV Routines

1: PHOSCH

2: P10D

3: P15BC

PVTURN Routines

1: P13A

2: P14A

3: P05

4: P03

-1 means further processing within PSPEC is required.

PVSP1 Routines

1: -1

2: P12C

3: -1

4: -1

5: -1

6: PMWAV

7: P15SCH

8: P15SCH

9: CKEZN

10: P15SCH

11: CKAGP

PVSP2 Routine: CKTLS

PVSP3 Routine: -1

PVPCH Routine: P02C

PMVC.CO, CPU 1 PMV Scores (Continued)

*The following table describes the bit settings in PVNEX:

Bit In PVNEX	Subroutine To be Called	PMV Interested	Reason Interested
2	AFDNA	P04	No "over" after do not acknowledge
3	AFAPGP	P04	"Over" may follow approaching glidepath
4	AFWC	P04	"Over" may follow wheel check
5	OLTCK	P11	"Over" or final course pos. follows OLT
6	DHCK	P09	"Over" may follow at decision height
7	PATCK	P12	"Button X clear" follows RO
8	WOCK	P15	Expect part of WO message
9	HOCK	P12	Need rest of handoff
10	PMCAM	P10,P15	Expect WO turns
15	***	***	Dummy bit stops PSUB(*) call

PRMPT.CO, CPU 1 Prompt information.

Phase routines help update this information. GLIB also uses this information.

Common Variable	Type	Description
PRDEV	I	Last prompt device for final controller 1 = $VRO 2 = CRT 3 = Audio
PRDNE	L	True, if something is being said by the votrax or digitized audio
PRQUE	IA	Array. Queue of messages to be said. First word is output device (negative value). Following words are phrase numbers.
PRENT	I	PRQUE entry position pointer
PRSAY	I	PRQUE start of message to be output pointer

PZ3CM.CO, CPU 1 PZ38 Common.

Mnemonic	Type	Content
P3S(1)	IA	% Starting position 1: minimum fuel
P3S(2)	IA	% Starting position 2: right base (cumulative)
P3S(3)	IA	% starting position 3: straight in (cumulative)
P3S(4)	I	% Starting position 4: left base (cumulative)
P3SV(1)	IA	% Heading variation on position 1 starts
P3SV(2)	IA	% Heading variation on position 2 starts (cumulative)
P3SV(3)	IA	% Heading variation on position 3 starts (cumulative)
P3SV(4)	IA	% Heading variation on position 4 starts (cumulative)
P3SP(1)	IA	% Slow aircraft
P3SP(2)	IA	% Medium aircraft (cumulative)
P3SP(3)	IA	% Fast aircraft (cumulative)
P3P(1)	IA	% Type 1 pilot (best)
P3P(2)	IA	% Type 2 pilot (cumulative)
P3P(3)	IA	% Type 3 pilot (cumulative)
P3P(4)	IA	% Type 4 pilot (cumulative)
P3P(5)	IA	% Type 5 pilot (worst)
P3A(1)	IA	% Full stop
P3A(2)	IA	% Low approach (cumulative)
P3A(3)	IA	% Touch & go (cumulative)
P3NGY(1)	IA	Percent no gyro approaches
P3NGY(2)	IA	Cumulative percent non no gyro approaches
P3CL(1)	IA	Percent clearance given at first request
P3CL(2)	IA	Cumulative percent continue then clear at 2 miles
P3CL(3)	IA	Cumulative percent no response
P3CL(4)	IA	Cumulative percent wave off
P3CL(5)	IA	Cumulative percent clearance given then cancelled
P3WN(1)	IA	Percent light and variable winds
P3WN(2)	IA	Cumulative percent 190 deg at 10 kts

PZ3CM.CO, CPU 1 PZ38 Common (Continued)

Mnemonic	Type	Content
P3WN(3)	IA	Cumulative percent 190 deg at 20 kts
P3WN(4)	IA	Cumulative percent 250 deg at 10 kts
P3WN(5)	IA	Cumulative percent 250 deg at 20 kts
P3LA(1)	IA	Percent low altitude alert conditions met
P3LA(2)	IA	Cumulative percent low altitude alert conditions not met
P3MS(1)	IA	Percent minimum separation violation runs
P3MS(2)	IA	Cumulative percent non minimum separation violations
P3ICE(1)	IA	Percent runs with icing
P3ICE(2)	IA	Cumulative percent no icing
P3HYF(1)	IA	Percent runs with hydraulic failure
P3HYF(2)	IA	Cumulative percent no hydraulic failure
P3ENG(1)	IA	Percent runs with single engine failure
P3ENG(2)	IA	Cumulative percent no engine failure
P3WL(1)	IA	Percent of runs with wheels down before controller asks.
P3WL(2)	IA	Cumulative percent of runs with wheels up
P3RV(1)	IA	Percent of runs in which ruway is visible at decision height.
P3RV(2)	IA	Cumulative % of runs in which runway is not visible.
P3WD	IA	Wind direction categories
P3G	IA	Gustiness categories
P3WS	IA	Wind speed categories

RDR.CO, CPU 1 Radar Information.

Common Variable	Type	Description
RDSVAZ	I	Servo position, azimuth antenna
RDSVEL	I	Servo position, elevation antenna
RDCLR	I	Centerline alignment zone
RDTDR	I	Touchdown alignment zone
RDRNG	I	Range alignment zone (concerns touchdown)
RDPLSZ	I	Not used
RDXALPO	I	Position of elevation servo during alignment
RDYALPO	I	Position of azimuth servo during alignment
RDALIM	I	Standard azimuth servo limit
RDELIM	I	Standard elevation servo limit
RDAMAX	I	Azimuth maximum display limit
RDAMIN	I	Azimuth minimum display limit
RDEMAX	I	Elevation maximum display limit
RDEMIN	I	Elevation minimum display limit
RDCHG	I	Display change to avoid burning phosphor on Megatek
RDAZR	L	True if azimuth radar on
RDELR	L	True if elevation radar on
RDAZS	L	True if azimuth servo on
RDELS	L	True if elevation servo on
RDAZN	I	Azimuth radar zone
RDEZN	I	Elevation radar zone
RDALT	I	Midpoint of elevation target before clipping
RDCRS	I	Midpoint of azimuth target before clipping
RDY1	I	Top coordinate of azimuth target
RDY2	I	Bottom coordinate of azimuth target
RDY3	I	Top coordinate of elevation target
RDY4	I	Bottom coordinate of elevation target
RDHALF	I	Halfsize of the target before clipping
RDAZNS	IA	Array of limits for each azimuth servo zone
RDEZNS	IA	Array of limits for each elevation servo zone
RDAZH	L	True if azimuth hashmarks are on
RDELH	L	True if elevation hashmarks are on
RDX1	I	Range coordinate of targets

RDR1.CO, Radar-related Parameters.

Parameter Name	Value	Description
RDAZHISLOPE	.216	High slope of outer azimuth scan
RDAZLOSLOPE	-.25	Low slope of outer azimuth scan
RDELHISLOPE	1.1	High slope of outer elevation scan
RDELOSLOPE	-.05	Low slope of outer elevation scan
RDAYEND	-1117	Radar point, azimuth (Y coordinate)
RDEYEND	-186	Radar point, elevation (Y coordinate)
RDXEND	-1655	Radar point on both (X coordinate)
RDXVALUE	1331	Result of subtraction of 1 mile coordinate and radar point (x coordinate)
RDWIDTH	267	Width of 1 mile hashmark
RDMEGAMAX	2047	Maximum of display area
RDMEGAMIN	-2047	Minimum of display area

RECKON.CO, CPU 2 Voice Recognition Common Block.

Common Variable	Type	Description
RCGPP	I	Correct glidepath position mask, or -1
RCGPT	I	Correct glidepath trend mask, or -1
RCCRP	I	Correct course position mask, or -1
RCCRT	I	Correct course trend mask, or -1
RCRNG	I	Correct range, including DH and OLT, or -1
RCOTHR	I	Other final controller masks, or -1 (All of above data initialized to -1)
RCEMERG	I	Emergency wave-offs, on -1
RCRES(7)	IA	Resolution masks for the above controller messages.
RCPHS(7)	IA	Phase of flight masks to be used as resolution masks. These mask values are data initialized.
RCBF(1)	IA	Recognition information buffer: word 1 =11, identifies this as a recognition block
RCBF(2)		Time of LP4 in .5 second ticks from the start of the problem
RCBF(3)		Time in 100 msec. ticks from above timer
RCBF(4)		First choice message recognized
RCBF(5)		Heading flag, or -1
RCBF(6)		Wind flag, or -1
RCBF(7)		Second recognition choice, or -1
RCBF(8)		Missed approach flag, or -1
RCFZIS	I	Phase of flight 1 = initial handoff 2 = body of approach 3 = final handoff
RCMSP	IA	Special flag masks
RCRSP	IA	Special resolution amasks

SBF.CO, Digitized Speech Buffer Area the Actual Buffers are Allocated by Loading a Correlative Low Level Routine, SDBF.SR.

Common Variable	Type	Description
SBF1	IA	Digitized speech buffer 1
SPF2	IA	Digitized speech buffer 2

Parameter Name	Value	Description
SBSIZE	1024	Size of hard wired digitized speech buffers

SHUSH.CO SUS Common.

Common Variable	Type	Description
SSBFA(1)	I	Buffer lock
		Bit 12-15-0: Buffer being filled
		Bit 15-1: Buffer ready
		Bit 14-1: APE release
		Bit 13-1: APE release
		Bit 12-1: Controller release
		Bits 11-8: Phrase concatenation count
		Bits 7-0: Phrase type
SSBFA(2)	I	Time of LP4 in .5 second ticks from the start of the problem
SSBFA(3)	I	Time in 100 msec. ticks from above
SSBFA(4)	I	First choice phrase understood
SSBFA(5)	I	Heading, if any (turns, wind) Missed approach position
SSBFA(6)	I	Call sign indicator, or wind speed Button # for missed approach
	I	Bits 13-15: Call sign 1-4
SSBFA(7)	I	Second choice phrase understood, or -1
SSBFA(8)	I	-1 if mike is keyed, else 0
SSBFB	I	As above
SSBFO(1)		Serves as input to activity replay file. Word 1 equals: -1, not in use
		1. Awaiting next phrase understood
		3. Ready to output
SSBFO (2-7)	I	As in SSBFA and SSBFB
SSBFO(8)	I	If bit cleared
		Bit 15: End of message
		Bit 14: Correction applied
		Bit 13: Over applied
		Bit 12: Improper SYNTAX
		Bit 11: Preliminary message
SSCAT(1)	I	Gives status for each phrase type task, Array (1:6)

SHUSH.CO SUS Common (Continued)

Common Variable	Type	Description
SSCAT(2)	I	Wind status, or -1
SSCAT(3)	I	Missed approach status, or -1
SSCAT(4)	I	Digit status, or -1
SSCAT(5)	I	Misrecognitions status, or -1
SSCAT(6)	I	Other status, or -1
SSBFW(8,6)	I	SUS working buffers. Same format as SSBFA and SSBFB for each of the above tasks
SSDIG(2,3)	I	Each entry is a digit to be stored. The digit task is the only one which actually fills it.
SSHEAD	I	Corrected course heading at time of student input provided by model controller
SSHDG	I	Aircraft heading at time of student input
SSRNG	I	Aircraft range at time of student input
SSBFI(1)	I	Array (1:8). SUS input buffer. Word 1 = 11, identifies this as a recognition block
SSBFI(2)	I	Time of LP4 in .5 second ticks
SSBFI(3)	I	Time of 100 msec. ticks from above time first choice phrase
SSBFI(4)	I	First choice phrase
SSBFI(5)	I	Heading flag, or -1
SSBFI(6)	I	Wind flag, or -1
SSBFI(7)	I	Second recognition, or -1
SSBFI(8)	I	Missed approach flag, or -1
SSMXS	IA	Array(6). Max value of corresponding entry in SSCAT
SSCOM	I	Array(9). First element is number of digit combinations available. Following entries are actual combinations
SSID	I	Array(NVRP). Data initialized to phrase ID
SSHFG	I	Heading advisory type
SSOHDG	I	Old heading from last correction

SHUSH.CO SUS Common (Continued)

Common Variable	Type	Description
SSUSE	I	Speech understood buffer use flag
SSIS1	I	ID of first choice recognition
SSIS2	I	ID of second choice recognition
SSYNTX	IA	SYNTAX map for all phrases. Array (0;111)
SSNXT	IA	SYNTAX phrase list. Array(48). Yields next phrase
SSAPEP	I	PTR. to SUS buffer to be processed by ape
SSMOD	I	PTR. to SUS buffer to be processed by model

Parameter Name	Value	Description
SSNB	2	Number of SUS buffers
SSCLN	84	Number of words to reset to -1; SSBFA-thru-SSDIG

SKED.CO, CPU 1 Scheduling Information.

Common Variable	Type	Description
SKTEN	IA	Linked list of tasks to be called with times
SKNXT	IA	Pointer to next task in time list
SKTIME	IA	Time for this task to be called
SKTNX	I	Next task to be called in SKTEN
SKTAV	I	Unused
SKRIX	IA	Index into CTMSG
SKMSG	IA	Message to be put in CT***
SKNXR	IA	Pointer to next task in range list
SKREN	IA	Linked list of tasks to be called, messages
SKRNG	IA	Array of ranges at which tasks in SKREN are to be called (miles*100)
SKRNX	I	Next task to be called
SKRAV	I	Unused

Parameter Name	Value	Description
SKNULL	32000	Used as end of list indicator
SKTASKNUM	50	Maximum number of routines schedulable
SKPUT	0	Used by RNGSCHD to indicate that this is not a routine to be executed.

SPACT.CO, Activity File Information, Speech and Aircraft Position, Actout Buffers and Pointers

Common Variable	Type	Description
SALOCK	I	Record type
SALP4	I	Time of LP4 in .5 second ticks from the start of the problem
SACLP4	I	Time in 100 msec ticks from above timer
SAFRST	I	First choice message understood (message actually given to APE).
SAHEAD	I	Heading, etc., if any.
SACS	I	Call sign, or wind speed
SASCND	I	Second choice message understood, or -1.
SACORCT	L	T if correction was applied to this phrase
SAGPP	I	Correct glidepath position message
SAGPT	I	Correct glidepath trend message, or -1
SACRP	I	Correct course position message
SACRT	I	Correct course trend message, or -1
SAOTHR	I	Other final controller messages
SAMIKE	L	T if mike keyed
SAZ	I	Azimuth zone
SAEZ	I	Elevation zone
SATR	I	Azimuth trend
SAETR	I	Elevation trend
SAMILE	I	Miles to touchdown *100
SAHDG	I	Aircraft heading
SAOHDG	I	Old heading, or what was given as heading last
SACSTYPE	I	Scoring type
		1 = Event detector record
		2 = Mile mark record
		3 = OLT, ADH, and mile report record
		4 = Wind record
		5 = No-gyro correction record
SARNTYP	I	Indicates type of event for SACSTYPE=1 and 3

SPACT.CO, Activity File Information, Speech and Aircraft Position, Actout Buffers and Pointers (Continued)

Common Variable	Type	Description
SAWHDG	I	Wind heading (when clearance given)
SAWSP	I	Wind speed (wind speed when clearance given)
SACTR	IA	Array to equivalence with model messages, etc.
SAABF	IA	SP1ACT: Equivalence to 'A' buffer
SABBF	IA	SP1ACT: Equivalence to 'B' buffer
SAIPT	I	SP1ACT: PTR. into current actout buffer
SAOPT	I	Pointer into current buffer, used by RDACT
SAOBN	I	Buffer number for double buffering, used by RDACT
SAIBN	I	SP1ACT: Actout buffer # for double buffering
SABUF	IA	SP1ACT: Double array for double buffering
SABLN	I	Block number read from RPLACT
SAMSG	IA	Array to equivalence with SUS messages
SADOA	I	DOA word from panel driver
SADOB	I	DOB word from panel driver
SA00A	I	Old DOA word
SA00B	I	Old DOB word
SANGZ	I	No-gyro zone 1/2 mi. after "HDG.xxx"
SANGT	I	No-gyro trend 1/2 mi. after "HDG.xxx"
SADIS	I	Azimuth display coord. for A/C center
SAVRO	I	Votrax phrase from automated voice record
SAMIFDG	I	Range schedule fudge factor allows for 2 second SUS fudge factor

Parameter Name	Value	Description
SA1CLR	25	Clear out first 25 words in SPACT
SA2CLR	4	Clear out 4 words beginning with SANGZ

SPCH.CO, CPU 2 Speech Associated Variables. Includes All Phrase Related Constant Data.

Common Variable	Type	Description
SPAT	IA	Each word in the array is associated with a final controller phrase for the current student. Each word acts as a flag for SUS and audio prompts. If 0 = No VRP 1 = VRP formed Array is in phrase ID order
SPVAL	IA	Array (1:N) in phrase ID order. Each entry yields the validated percentages for the corresponding phrase VRP. It is set to zero before validation occurs.
SPDEV	I	Prompt device last used. 1 = $VRO 2 = CRT 3 = Audio
SPFLG	I	<0 =VDC, 0 = none, >0 = SUS on
SPLVL	I	Voice input level
SPID	IA	Array (1:N) is in phrase ID order. Each entry is the phrase ID.
SPVRP	IA	VRP file position pointer. Array is in phrase ID order. Each entry is a pointer to the start of the VRP
SPNUM	IA	Array (1:N) is in phrase ID order SPNUM (1:N) indicates # of IFPS necessary to form a VRP
SPLST	IA	Array (1:7).SPLST (1) may indicate either % validation or a phrase number. SPLST (2-7) are either phrase numbers or set to zero.
SPIFP	IA	Array is in phrase ID order. File position pointers to IFP storage location.

SPDGT.CO, CPU1 Digitized Speech Variables.

Common Variable	Type	Description
SDFRAZIN	I	Presently active phrase number recording
SDFRAZOUT	I	Presently active phrase number playing
SDCURBUF	I	Presently inactive buffer
SDFLG	I	Present mode of digitizer (1-7) 1-Inactive 2-Record 3-Play 4-Record/play 5-Start play at next interrupt 6-Buffer in anticipation of switch to play 7-Start record at next interrupt
SDCHN	I	Channel opened for requested phrase
SDRECSTRT	I	Starting address of record block
SDRECEND	I	Ending address of record block
SDPLAYSTRT	I	Starting address of playback
SDPLAYEND	I	Ending address of playback
SDCANINDX	IA	Holds indexing addresses of canned phrases
SDWRNEXT	I	Address of next available record (write) location

Parameter Name	Value	Description
SDBUFBLOK	2	Beginning speech buffer window block
SDUSRBLOK	0	Beginning of user space window block
SDCFRN	27	Number of canned phrases

SPEECHMESSAGES.CO. This is the parameter block which contains all speech messages which a trainee may encounter during voice data collection and validation.

Parameter Name	Value
SMTIMEOUTMSG	"Can't hear you! <15> *Voice level adjusted? <15> *Super key deselected?"
SMTRYAGAINMSG	"Ready for another try? <15> Hit any key to continue."
SMNOTKNOWNMSG	"Huh....? <15> Your input was not understandable."
SMTOOSHORTMSG	"Your input was too short! <15> *Mike and input level adjusted? <15> *Said too quickly? <15> *Hiccup?"
SMTOOLONGMSG	"Your input was too long! <15> *Forgotten pause? <15> *Too slow?"
SMLOWRECMSG	"The phrase has been recognized at a low confidence level."
SMWRONGMSG	"The wrong phrase has been recognized."
SMGOODMSG	"Good phrase input."
SMPROMPTMSG	"<FF>Repeat the following phrase(s):"
SMSAYSOMEMSG	"<FF>Say any phrase that you have learned. <15> See if I can recognize them."
SMLOWLEVELMSG	"Your voice level is low <15> *Voice level adjusted? <15> *Mike positioned correctly?"
SMSTYLEMSG	"Too many phrases! <15> *Extra pauses? <15> *Cough?"
SMMISSMSG	"Too few phrases! <15> *Pauses not long enough? <15> *Forgotten pause?"
SMYOUSAIDMSG	"Your input was recognized as:"
SMSTOPMSG	"Validation not successfully completed."
SMENDMSG	"Validation voice test run terminated."

SRV.CO, CPU 2 Servo Information.

Mnemonic	Type	Content
SVLOX	I	Servo limit, lower X
SVLOY	I	Servo limit, lower Y
SVHIX	I	Servo limit, upper X
SVHIY	I	Servo limit, upper Y
SVSETX	I	Location of servo, X-plane
SVSETY	I	Location of servo, Y-plane
SVZN	IA	Displacement to add to reflector position
SVSTDA	IA	Standard position azimuth touchdown reflector
SVSTDE	IA	Standard position elevation touchdown reflector
SVMSG	I	Message code for servo routine, 1-azimuth change. 2-Elevation change, 3-freeze azimuth, 4-freeze elevation, 5-Move hashmarks without moving servo
SVY	I	Servo position on either X or Y plane
SVX	I	Position of reflectors dependent on SVMSG for which one
SVSCL	IA	Standard position centerline reflector
SVTDAREF	I	Reference for azimuth touchdown reflector
SVTDEREF	I	Reference for elevation touchdown reflector
SVSCLREF	I	Reference for centerline reflector
SVAM	R	Servo midpoint changes for azimuth
SVEM	R	Servo midpoint changes for elevation
SVXPOS	I	Old position of X servo
SVYPOS	I	Old position of Y servo
SVXNEW	I	New position of X servo
SVYNEW	I	New position of Y servo

SRV.CO, CPU 2 Servo Information (Continued)

Parameter Name	Value	Description
SVAZMAX	-282	Azimuth maximum servo
SVAZMIN	-1450	Azimuth minimum servo
SVELMAX	628	Elevation maximum servo
SVELMIN	-186	Elevation minimum servo
SVAMID	-978	Servo midpoint azimuth
SVEMID	-2	Servo midpoint elevation
SVAZCLR	1	Change azimuth servo, and centerline reflector
SVAZTDR	2	Change azimuth servo and touchdown reflector
SVETDR	3	Change elevation servo and touchdown reflector
SVAZFRZ	4	Freeze azimuth servo
SVEFRZ	5	Freeze elevation servo
SVMOVE	6	Move hashmarks without moving servo, i.e., replay
SVGRDT1	7	Move hashmarks during real run
SVSRMON	8	Move hashmarks where SRMON says to

SRV1.CO, Parameters for the Servo.

Parameter Name	Value	Description
SVAZMAX	-282	Azimuth maximum servo
SVAZMIN	-1450	Azimuth minimum servo
SVELMAX	628	Elevation maximum servo
SVELMIN	-186	Elevation minimum servo
SVAMID	-978	Servo midpoint azimuth
SVEMID	-2	Servo midpoint elevation

SUSAY.CO. This is a CPU 1 comon block for the ISAY environmental information buffer area.

Common Variable	Type	Description
SUBUF	IA	Holds environmental information until it can be appended to the appropriate SUS buffer for output to the student activity file
SUFRST	I	Pointer to first buffer segment which is to be appended.
SUNTRY	I	Pointer to segment which is to be filled next.

Parameter Name	Value	Description
SUFIND	1	Wants to find buffer for appending
SUSTORE	2	Wants to store environmental info
SUTRM	3	Wants most recent record
SUSTART	1	Starting index of buffer array
SUSEG	9	Size of each buffer segment
SUNUM	5	Number of segments
SUEND	SUSEG*SUNUM	Buffer array end index

TEXT2.CO, CPU 2 Text Presentation Information used by PLATEXT

Common Variable	Type	Description
TEXT2	IA	Non-decoded text presentation info from P1TXT in CPU 1. The format is as described in Appendix F of the GCA-CTS design report.

TZC.CO, CPU 1 TZEC Common.

Common Variable	Type	Description
TZCSYL	IA	Syllabus file CHSAV
TZCSR	IA	Student record file CHSAV
TZPHZ	IA	Problem file CHSAV
TZP3	I	Pointer to available phase 3 problem block
TZSR1	I	Record # of available block in student file #1
TZPV19	I	Pointer to the next PV19 score
TZSUM	I	Record # of the next available phase 3 summary block
TZLSUM	I	Record # of the last phase 3 summary block

VICOM.CO, Voice Input Communication Block for CPU 2.

Common Variable	Type	Description
VIBF1	I	Buffer 1 start address
VIBF2	I	Buffer 2 start address
VIEND	I	Buffer 1 end
VIUSE	I	Buffer use flag
VI500	I	LP4 time in half seconds
VI100	I	LP4 time in 100 millisecond offsets
VIBFA	I	Buffer A flag. Set to relative window block when in use
VIBFB	I	Buffer B flag

VINA1.CO. This is the CPU 2 speech buffer common. These buffers are window mapped and thus must be loaded at a 1K word boundary.

NOTE: VINA1.CO contains common blocks VINA1, VINA2, VINB1, VINB2.

Mnemonic	Type	Content
INA1	IA	First half of speech buffer A
INA2	IA	Second half of speech buffer A
INB1	IA	First half of speech buffer B
INB2	IA	Second half of speech buffer B

Parameter Name	Value	Description
BFSZ	2048	Allows 4 sec. of speech
WNSZ	8	8 window blocks
BLKCNT	23	Number of 256 word blks

VLID.CO, Validation Variables.

Common Variable	Type	Description
VLPCT	I	Validation percentage requested
VLSTF	L	True = stifle requested by student
VLARGS	IA	Phrases to be validated.
VLTOC	I	User clock counter

VOCIN.CO. VOCIN holds information which is necessary to voice data collection for the present student.

Common Variable	Type	Description
VCFRAZ	IA	Array in phrase ID order. Array (1:NVRP) is # of IFPS ready (maximum is 4 or 10)
VCSLOT	IA	Array (1:NVRP) is LFP slot to be used (between 1-4 or 1-10)
VCWGT	IA	Array initialized to number of times features must be set given number of repetitions (0,0,0,1,1,1,2,2,2,3)

VSC01.CO, Votrax Variables.

Common Variable	Type	Description
IFIL	I	Bumped by VSCON every time word inserted in IPHRZ
IOLD	I	Bumped every time word is output from IPHRZ
IPHRZ	IA	Circular array which holds phrases to be output

VSIFP.CO, CPU 2 Storage of Voice Input Data and Constants Associated with IFP Formation and Recognition Data.

Common Variable	Type	Description
VSBF1	I	Buffer 1 start address
VSBF2	I	Buffer 2 start address
VSEND	I	Buffer 1 end address
VSUSE	I	Buffer use flag
VS500	I	LP4 time in half seconds
VS100	I	LP4 time in 100 millisecond offsets
VSMSG	I	Driver message flag
VSTIM	I	Sample length of input features
VSTSL	I	Number of time slots
VSM	I	Minimum score for any decision
VSIP1	I	PTR. to storage for 32 slot IFP
VSIP2	I	PTR. to storage for 16 slot IFP
VSIWBLK	I	Relative window block # for present buffer start
VSLOCT	I	PTR. to locator table
VSCADR	I	PTR. to score area
VSCAR	IA	Array (1:32). Feature count array
VSNM1	IA	Normalization scores for short IFP
		1-Unshifted
		2-Shifted, lost last slot
		3-Shifted, lost first slot
VSNM2	IA	Normalization scores for long IFP
		1-Unshifted
		2-Shifted, lost last slot
		3-Shifted, lost first slot
VSCOMT	I	PTR. to VRP being examined
VSIF	I	PTR. to IFP to be used
VSWDNO	I	Block # presently in window
VSNMAL	I	PTR. to normalization factor to be used.
VSCHOT	IA	Choice table

VSIFP.CO, CPU 2 Storage of Voice Input Data and Constants Associated with IFP Formation and Recognition Data. (Continued)

Parameter Name	Value	Description
VST	10	Closeness factor-used in VCHOS
VSCLS	10	Breaux test closeness factor
VSTV	10	Breaux test multiplier value
VSCNF	10	VALYZ closeness factor
VSMDFLT	40	Default minimum score
VSNOCHS	2	Number of choices
VSNWBLK	3	Maximum # of window blocks to be used for VRPS per half window
VSWSIZE	104K	3K data window (140 octal 32 word rec)
VSWNMX	8	Total # of 1K blocks in window
VSHRT	500	500/VSTSL added in VALYZ if short VRP
VSLNG	200	VSTSL/200 added if long VRP
VSFRV	9	First VRP record #
VSLWIN	1	Last window block set
VSLBLK	3	# of blocks in last set
VSHFL5	5	Shifts required to multiply by VRP file record size (32 words)

VX.CO, Parameter List for Use by Votrax and SUS.

Parameter Name	Value	Description
VX1MI	1	1 mile
VX1HM	2	1 and 1/2 mile
VX2MI	3	2 miles
VX2HM	4	2 and 1/2 miles
VX3MI	5	3 miles
VX3HM	6	3 and 1/2 miles
VXAT	7	At
VX12	8	12
VX15	9	15
VX20	10	20
VX25	11	25
VX30	12	30
VX0	13	0
VX1	14	1
VX2	15	2
VX3	16	3
VX4	17	4
VX5	18	5
VX6	19	6
VX7	20	7
VX8	21	8
VX9	22	9
VXCTW	23	Contact tower after landing
VXB1C	24	Button 1, clear
VXB2C	25	Button 2, clear
VXMA	26	Missed approach
VXRNI	27	If runway not in sight
VXERA	28	If runway not in sight, execute missed approach
VXIMF	29	If runway not in sight, climb and maintain 1500
VXBT1	30	Button 1

VX.CO, Parameter List for Use by Votrax and SUS (Continued)

Parameter Name	Value	Description
VXPHO	31	Proceed direct point bravo; hold until advised by GCA
VXBT2	32	Button 2
VXONT	33	On the go
VXOLT	34	Over landing threshold
VXTLS	35	Too far left for safe approach
VXTRS	36	Too far right for safe approach
VXOCE	37	On centerline
VXLCE	38	Left of centerline
VXSLE	39	Slightly left of centerline
VXRCE	40	Right of centerline
VXSRE	41	Slightly right of centerline
VXTLO	42	Too low for safe approach
VXTHS	43	Too high for safe approach
VXWND	44	Wind
VXCFL	45	Cleared for low approach
VXCFT	46	Cleared for touch and go
VXCTL	47	Cleared to land
VX1MT	48	1 mile from touchdown
VX2MT	49	2 miles from touchdown
VX3MT	50	3 miles from touchdown
VX4MT	51	4 miles from touchdown
VXWLO	52	Well left of course
VXLCO	53	Left of course
VXWRO	54	Well right of course
VXRCO	55	Right of course
VXWBG	56	Well below glidepath
VXWAG	57	Well above glidepath
VXFBG	58	Going further below glidepath
VXFAG	59	Going further above glidepath
VXCMF	60	Climb and maintain 1500

VX.CO, Parameter List for Use by Votrax and SUS (Continued)

Parameter Name	Value	Description
VXADH	61	At decision height
VXOCO	62	On course
VXSLO	63	Slightly left of course
VXSRO	64	Slightly right of course
VXCNG	65	Correcting
VXOGL	66	On glidepath
VXBGP	67	Below glidepath
VXSGB	68	Slightly below glidepath
VXAGP	69	Above glidepath
VXSAG	70	Slightly above glidepath
VXGBG	71	Going below glidepath
VXCUP	72	Coming up
VXGAG	73	Going above glidepath
VXCDO	74	Coming down
VXP4R	75	Position 4 roger
VXRB1	76	Radar button one
VXRB2	77	Radar button two
VXTFC	78	This is your final controller, how do you hear me?
VXWSD	79	Wheels should be down
VXDNA	80	Do not acknowledge further transmissions
VXAPG	81	Approaching glidepath
VXBGD	82	Begin descent
VXGB1	83	Give me button 1
VXGB2	84	Give me button 2
VXARM	85	Army 876
VXMAR	86	Marine 687
VXNAV	87	Navy 310
VXAF	88	Airforce 307
VXOVR	89	Over
VXTNG	90	This will be a no-gyro PAR approach

VX.CO, Parameter List for Use by Votrax and SUS (Continued)

Parameter Name	Value	Description
VXMHS	91	Make half standard rate turns
VX5MT	92	5 miles from touchdown
VX6MT	93	6 miles from touchdown
VX7MT	94	7 miles from touchdown
VX8MT	95	8 miles from touchdown
VXLAL	96	Low altitude alert, check your altitude immediately
VXHWN	97	How do you hear me now?
VXCTN	98	Correction
VXTRI	99	Turn right
VXSTT	100	Stop turn
VXTLE	101	Turn left
VXXMS	102	Execute missed approach
VXRCL	103	Radar contact lost
VXCMT	104	Climb and maintain 3000
VXTRH	105	Turn right heading
VXHED	106	Heading
VXTLH	107	Turn left heading
VXACL	108	After completing
VXCLS	109	Contact lost
VXDYC	110	Did you copy?
VXFS	111	Full stop
VXHO	112	Hand off
VXLB	113	Left base
VXLA	114	Low approach
VXMF	115	Minimum fuel
VXP3	116	P3
VXPOS	117	Position
VXRC	118	Radar contact
VXRB	119	Right base
VXSIN	120	Straight in

VX.CO, Parameter List for Use by Votrax and SUS (Continued)

Parameter Name	Value	Description
VXT38	121	T38
VXTAG	122	Touch and go
VXU21	123	U21
VXWI	124	Where is
VXYMA	125	Your missed approach procedure is
VXA6	126	A6
VXWBC	127	Weak but clear
VXLAC	128	Loud and clear
VXWDL	129	Wheels down and locked
VXMAP	130	Missed approach
VXRIS	131	Runway in sight
VXOUT	132	Out
VXROG	133	Roger
VXPTL	134	Turn left heading (pilot)
VXPTR	135	Turn right heading (pilot)
VXPTH	136	Heading (pilot)
VXSHS	137	Should have said
VXPAS	138	(Pause)
VXDTS	139	Don't touch the servo
VXDIG	140	(Digits follow)
VXVAR	-1	(Variable no. of phrases follow)
VXBUF	0	(Address of buffer follows)
VXKIL	500	(Kill Votrax)
VXMAX	141	(Maximum phrase no.)

XPOSE.CO, Common Block for Images.

Common Variable	Type	Description
XPAZ	RA	Values for azimuth display coordinates
XPEL	RA	Values for elevation display coordinates
XPAHSH	RA	Values for hashmark coordinates on azimuth
XPEHSH	RA	Values for hashmark coordinates on elevation
XPRNG	RA	Values for range coordinates
XPTDCHK	RA	Values for touchdown check line on elevation
XPDECHT	RA	Values for decision HT line

APPENDIX C

FILE STRUCTURES

This appendix describes the format of the files used by GCA-CTS. The file structures contained in this appendix are as follows:

Syllabus File

The syllabus file guides the entire sequence of GCA-CTS training. It is a card image file containing the names of the individual phase instruction files, thus there is an entry for each phase of each task as well as for the performance test. Since not all phases are included in each task, the phase to which each file relates is identified. Position within this file is maintained by storing CHSAV information in the student file. File format is:

Column	Content	Meaning
1	C	Comment
	1	Phase 1 instruction file
	2	Phase 2 instruction file
	3	Phase 3 instruction file
	4	P-run file
3-15	filename	Name of instruction file

Example:

```
C TASK 1.1
1 T01$01.01
C TASK 1.2
1 T01$02.01
2 T01$02.02
```

Phase 1 Instruction File

Phase 1 teaches the proper use of radio terminology while formulating the student's voice reference patterns. Creative utilization of the terminal display, graphics display, student panel, digitized audio, and voice synthesizer provide a variety of audio-visual aids to implement the phase 1 tasks. For each task a card image file defines the student voice data collection and instruction sequence. The file name identifies both task and phase, e.g., T01$03.01 is the task 1.3, phase 1 file. Instruction file names must be of the format: Taa$bb.0c

where aa = two digit number representing level
bb = two digit number representing problem
c = phase #

for example: T02$11.01

Nine types of cards may be used: comments, voice data collection instructions, display instructions, prompts, radar simulation instructions, aircraft simulation instructions, wait for events instructions, task file sequence instructions, and text display instructions. The eight latter card types are further divided into function card types. An instruction card generally contains the instruction type code in column 1 and the function identifier right-justified in columns 3 and 4, followed by the appropriate function arguments. All phase 1 instruction files are 80 byte per record files in which the record number and <CR> appear in the last four bytes. Detailed format descriptions of each card type follow.

Comments

Comment cards are defined as follows:

Column	Content	Meaning
1	C	Comment
3-79	comment text	

Voice Data Collection Instructions

Voice Data Collection (VDC) incorporates collection of input feature patterns (IFPs), formulation of voice reference patterns (VRPs) for reference array storage, and validation of collected VRPs (matching VRPs to student inputs). IFP collection is limited to the storage of the ten most recently obtained patterns. Upon request, these IFPs are used to formulate VRPs which are stored in the reference array for the speech recognition base. Provisions are made for six VDC functions as follows:

Instruction Type (Column 1)	Identifier (Column 4)	Arguments	Function
VDC = 1	1		Start VDC. Must be the first VDC function requested.
1	2	Columns 6-8=Phrase #s 10-12= . . 26-28=	Collect IFP for phrase(s). Does not provide a prompt. Waits for phrase(s) input.
1	3	Columns 6-8=Phrase #s 10-12= . . 26-28=	Form VRP for phrase. Necessary IFPs must be collected before this function is made valid.
1	4	Columns 7-8=% Accuracy 10-12=Phrase #s . . 32-30=	Validate to % accuracy, # phrases. Issues a prompt and validates student response. Continues until required accuracy is attained. Prompt is issued by most recent prompting device.
1	5		Validate, no prompting. Student may voice any phrase(s) learned thus far. The recognized phrase(s) are echoed.
1	6		Terminate VDC. All VRPs for which IFPs have been collected must be completed before this function is made valid.

Display Instructions

The graphics display is controlled and directed by the 10 functions defined in the table below. Graphic images, referred to as pictures, may contain a particular symbol, text segment, background display, etc. All active pictures create the final display image.

Instruction Type (Column 1)	Identifier (Column 4)	Arguments	Function
Display = 2	0	Columns 6-8 = XXX	Turn picture XXX off
2	1	Columns 6-8 = XXX	Turn picture XXX on
2	2	Column 8 = 0	Turn off all pictures and reset display
2	4	Column 8 = 0	Update aircraft
2	5	Column 8 = 0	Start display processor
2	6	Column 8 = 0	Stop display processor
2	7	Columns 5-79 = string	Type string
2	8	Column 8-0, or 1	Fade trails, 0-away, 1-to long trails
2	9	Columns 6-8 = XXX 10-12 = YYY	XXX=wind speed update in knots YYY=wind direction update in degrees
2	10	Column 8 = 0	Initialize display

Picture numbers are as follows:

1. Azimuth Display

2. Azimuth Hashmarks

3. Elevation Display

4. Elevation Hashmarks

8. Azimuth Target

9. Azimuth Trail

10. Elevation Target

11. Elevation Trail

12. Text String

13. Wind Advisories

14. Azimuth Long Trail

15. Elevation Long Trail

16. Touchdown Reflector on Elevation Picture

17. Touchdown Reflector and Centerline Reflector on Azimuth Picture

Prompts

Prompts may be issued via the CRT terminal display, the digitized audio, and the Votrax voice synthesizer. The model controller also may be called upon to give controller message prompts. This instruction may be used to modify the student panel status.

Instruction Type (Column 1)	Identifier (Column 4)	Arguments	Function
Prompts = 3	1	Columns 6-8 = XXX_1 ⋮ 26-28=XXX_6	Concatenate the phrases indicated and output on VOTRAX
3	2	Columns 6-8 = XXX_6 ⋮ 26-28=XXX_1	Concatenate the phrases indicated and output on the student CRT
3	3	Columns 6-8 = XXX_1 ⋮ 26-28=XXX_6	Concatenate the phrases indicated and output on the digitized audio device.
3	4	Column 8 = X, prompt device # 1-Votrax 2-CRT 3-Audio	Activate model controller activity using specified device(s) for output
3	5		Terminate model controller and APE activity
	6	Columns 6-8 = XXX phrase #	Store audio input for given phrase
3	7	Columns 7-8 = XXX_1 11-12 = YY_1 15-16 = XX_2	Cause the indicated changes to the student panel XX=panel device # (See Table C3) YY= -1 to turn on device = 0 to turn off device

Aircraft Simulation Instr[illegible]

Aircraft, pilot, and envir[illegible] [illegible] and activated by this grou[illegible] [illegible] is a set of two cards [illegible]

Instruction (Column 1)	Identifier (Column 4)	
4	1	APE initializat[illegible] content)
	2	Begin aircraft dynamics
	3	Freeze aircraft dynamics
	4	Terminate APE and model controller if it is running

APE Initialization Card Format

Card 1:

Column	Content	Meaning
1	4	Aircraft simulation instruction set
4	1	First card of an environmental simulation set
9		Type of flight:
	1	Pilot responds normally to controller advisories
	2	Restrict A/C position to contiguous azimuth zones given in columns 10-15
	3	Restrict A/C position to contiguous elevation zones given in columns 16-21
10-12		Left azimuth zone (e.g., -2)
13-15		Right azimuth zone (e.g., 1)
16-18		Lower elevation zone (e.g., -2)
19-21		Upper elevation zone (e.g., 1)
24		A/C type

Column	Content	Meaning
27		Pilot type: 1 = best, 5 = worst
30		Indicates the form of the information in columns
	1	A/C starting position specified in relation to environmental simulation, i.e., in feet and miles
	2	A/C starting position specified in relation to display simulation, i.e., by zone
31-35		A/C starting altitude in feet or by zone (right justified)
36-40		A/C starting offset in miles (xx.x) or by zone (right justified)
41-45		Starting range from touchdown in miles
46-50		Ending range in radar miles
51-55		Initial heading
77-79		Record number
Card 2:		
1	*	Continuation card
4		Approach type, used for the handoff message as follows:
	1	Full stop
	2	Low approach
	3	Touch-and-go
	4	Minimum fuel
	5	No-gyro
6		Clearance information for tower simulation:
	1	Clearance given at first request
	2	Continue, then clear at 2 miles
	3	Not given (no response)
	4	Wave off
	5	Clearance given then cancelled

Column	Content	Meaning
7-11		Number of seconds pilot will wait without radio contact before executing a missed approach
13	T,F	If T, pilot has wheels down before controller says "wheels down"
15	T,F	If T, cause A/C to descend to a point which requires a low altitude alert be given once during the run
17	T,F	If T, cause a gyro failure after handoff, at the point specified in columns 61-64
19	T,F	If T, a handoff is to be given
		Wind information:
20-24		Mean wind heading
25-29		Mean wind speed in knots
30-34		Mean gust speed in knots
35-39	xx.x	Mean gust duration in seconds
40-44	.xx	Fraction of time gusting occurs
45-49	x.xx	Wind variability (0.00 - 1.00)
50-54	xx.x	Wind speed correlation time
55-60		Ceiling height in feet
61-64	xx.x	Range from touchdown in miles when gyro failure occurs
77-79		Record number

Radar Simulation Instructions

Azimuth and elevation servo alignment and position are the primary objects of the radar simulation instructions. Azimuth and elevation servo angles are expressed as zones with respect to the glideslope and extended runway centerline. The azimuth servo angle zones range from -2 to 2 with 0 representing the alignment of the azimuth servo with the glideslope. The angle the glideslope forms with the ground is bisected to produce the zones -1 and -2. Likewise, zones 1 and 2 are produced by bisecting the angle made by

the glideslope and the upper limit of the azimuth servo angle. Elevation servo angles are similarly defined with the angle formed by the extended runway centerline and azimuth normal line being zone -1, alignment with the runway parallel line being zone 0, and the remaining 15° of azimuth sweep angle bisected being zones 1 and 2.

Servo manipulation includes activation of the joystick monitor and alignment changes. These alignment changes will vary from zone 0 (aligned) to zone 3 (very badly aligned). Azimuth radar scan limits are reflected in the hash-marks on the elevation display and vice versa. In the arguments shown below, if X = 32000, alignment position will not be changed. If Y = 32000, servo position will not be changed.

Instruction Type (Column 1)	Identifier (Column 4)	Arguments	Function
Servo = 5	1	Columns 6-10 = X* Columns 12-16 = Y**	Activate elevation radar servo, change alignment of centerline reflector to zone X, position servo at Y in X-plane.
5	2	Columns 6-10 = X* Columns 12-16 - Y**	Activate elevation radar servo, change alignment of range (affecting touchdown reflector) to zone X, position servo at Y in X-plane.
5	3	Columns 6-10 = X* Columns 12-16 = Y**	Activate azimuth radar servo, change alignment of touchdown reflector to zone X, position servo at Y in Y-plane.
5	4		Freeze elevation radar servo, change alignment of reflectors to 0 on both displays.
5	5		Freeze azimuth radar servo, change alignment of reflectors to 0 on both displays.

* if X = 32000, reflector position not changed
**if Y = 32000, servo position not changed

Instruction Type (Column 1)	Identifier (Column 4)	Arguments	Function
5	6	Columns 6-10 = X Columns 12-16 = Y X = elevation radar servo zone Y = azimuth radar servo zone	Change hashmark position to location which would be appropriate had the servo been moved to position (X,Y). If the student controller moves the servo out at this time, the hashmarks will not move.
Wait = 6	1	Columns 6-8 = XXX seconds	Delay xxx seconds before next operation.
6	2	Columns 6-8 = XXX, time-out specification in seconds 9-12=±XXX, skip # of cards on timeout, 14-15=XX_1, special key # 16-19=±XXX, skip # of cards on entry of XX_1. 49-50-XX_6, 51-54=±XXX, skip # of cards on entry of XX_6.	Wait for keyboard entry of special keys xx_1,..., or xx_6. (Special key definitions and codes are in Table C1).
6	3	Columns 6-8 = XXX, timeout specification in seconds. 9-12=±XXX, skip # of cards on timeout. 14-X_1, 15-18=±XXX, skip # of cards on entry of character X_1 . . 44=X_6 45-48=±XXX, skip # of cards on entry of character X_6	Wait for entry of standard keyboard characters.

Instruction Type (Column 1)	Identifier (Column 4)	Arguments	Function
6	4	Columns 6-8 = XXX, timeout specification in seconds 9-12=±XXX, skip # of cards on timeout 14-15=±X, angle zone	Wait for elevation servo angle zone (appears on azimuth display), where: $-2 \leq x \leq 2$
6	5	Columns 6-8 = XXX, timeout specification in seconds 9-12=±XXX, skip cards on timeout 14-15=±X, angle zone	Wait for azimuth servo angle zone (appears on elevation display), where: $-1 \leq x \leq 2$
6	6	Columns 6-8 = XXX, timeout specification in seconds 9-12=±XXX, skip # of cards on timeout 14-15=±X, azimuth zone	Wait for aircraft azimuth zone, where: $-3 \leq x \leq 3$
6	7	Columns 6-8 = XXX, timeout specification in seconds 9-12=±XXX, skip # cards on timeout 14-15=±XX, elevation zone	Wait for aircraft elevation zone, where: $-3 \leq x \leq 3$
6	8	Columns 6-8 = XXX, timeout specification in seconds 9-12=±XXX, skip # of cards on timeout 14-15=XX, range x 10	Wait for aircraft range from touchdown, xx in miles x 10

Instruction Type (Column 1)	Identifier (Column 4)	Arguments	Function
6	9	Columns 6-8 = XXX, timeout specification in seconds 9-12=±XXX, skip # of cards on timeout	Wait for Votrax to finish speaking
6	10	Columns 6-8 = XXX, timeout specification in seconds 9-12=±XXX, skip # of cards on timeout	Wait for end of digitized voice utterance
6	11	Columns 6-8 = XXX, timeout specification in seconds 9-12=±XXX, skip # of cards on timeout 14-15=XX panel input 18-19=YY	Wait for change in student panel input=xx, where panel inputs are described in Table C2. YY = 0 deselect; YY = -1 select
6	12	Columns 6-8 = XXX, timeout specification in seconds 9-12=±XXX, skip # cards on timeout	Wait for end of student voice input

Task File Sequence Commands

Normal sequencing through the phase 1 task file consists of a sequential card by card process with the exception of skip on timeouts and key entry discussed in the previous section. The processing sequence may be altered by utilization of skips, subroutine calls and returns, and conditional skips provided by the command formats listed below.

Instruction Type (Column 1)	Identifier (Column 4)	Arguments	Function
Sequence =7	1	Columns 5-8 = ±XXX	Skip # of cards from present record
7	2	Columns 6-8 = XXX, entry record # 10-13=±XXX_1 15-18=±XXX_2 ⋮ 30-33=±XXX_5 skip # of cards from present record number	Subroutine call with provisions for abnormal returns. Five levels of nesting are allowed.
7	3	Columns 7-8 = ±X, subroutine return where: X=0: return to card following subroutine call X=1,...,5: return to corresponding abnormal return point specified in previous subroutine call.	Subroutine return
7	4	Columns 7-8 = XX, flag # (1-10) 12-13=XX, condition #	Set flag to condition #
7	5	Columns 7-8 = XX, flag # (1-10) 12-13=XX, condition # 15-18=±XXX, skip XXX records from present record	If flag is set to the given condition #, the skip is
7	6	Columns 7-8 = -1	Return to start of file

Text File Display Commands

The phase 1 executive determines the text file name based upon the training file name. Thus text associated with task T01$20.01 is found in the file TX01$20.01. These text files are formatted into logical pages. The first of the following commands causes textual presentations to be retrieved from the text file. The other two allow text strings specified in the training file itself to be displayed.

Instruction Type (Column 1)	Identifier (Column 4)	Arguments	Function
Text = 8	1	Columns 6-8 = XXX	Logical page of text file to be presented
8	2	Columns 5-79 = text	Message to be typed on student's CRT
8	3	Columns 5-79 = text	Message to be typed on instructor's CRT

Table C1. Codes for Special Keys at Trainee Station

Key Code	Key Name
1	MENU
2	HELP
3	INIT VOICE TEST
4	STOP VOICE TEST
5	ALIGN
6	NEXT
7	YES
8	NO
9	Blank
10	HELLO
11	BYE
12	INIT NEW R/T

Table C2. Student Panel Wait States

(Used in wait conditions in phase 1 instruction files)

Identifier	Description
1	ICS button 3 is selected
2	ICS button 5 is selected
3	ICS button 7 is selected
4	ICS button SUPER is selected
5	Radio frequency 270.8 is selected
6	Radio frequency 318.8 is selected
7	Radio monitor 270.8 is selected
8	Radio monitor 318.8 is selected
9	Clearance is selected
10	Microphone is keyed
11	Not used
12	Clearance light is on
13	Waveoff light is on
14	Waveoff button is depressed
15	Student is speaking
16	Votrax is speaking

Table C3. Student Panel Device Codes

The following parameters are used in phase 1 with instruction 3 and 7 to effect a change in the student panel status.

Input Parameter	Panel Device Effected
0	ICS button "3" (amber)
1	ICS button "3" flashing (amber)
2	ICS button "5" (amber)
3	ICS button "5" flashing (amber)
4	ICS button "7" (amber)
5	ICS button "7" flashing (amber)
6	ICS button "SUPER" (amber)
7	ICS button "SUPER" flahsing (amber)
8	Radio frequency button "270.8" (amber)
9	Radio frequency button "270.8" (green)
10	Radio frequency button "318.8" (amber)
11	Radio frequency button "318.8" (green)
12	Alarm
13	Radio monitor button "270.8" (amber)
14	Radio monitor button "318.8" (amber)
15	Request clearance button (white)
16	Clearance received light (green)
17	Waveoff (red)
18	Instructor panel ICS (amber)
19	Instructor panel ICS flahsing (amber)

Phase 2 Problem Specification File

In phase 2, the system freezes and gives feedback to the student whenever an error is made on the new material. One phase 2 instruction file exists for every task which provides phase 2 training. It is a card image file whose suggested name incorporates both task and phase of instruction, e.g., for task 3.2 the file name should be T03$02.02. The file contains three types of cards: comments, the header information card, and environmental simulation sets (two cards each). The file contains exactly one header information card, and this preceeds any environmental simulation sets. In general, the header information pertains to all problems while the environmental simulation sets describe a particular problem. Card formats are described in the tabular form below. The code in column one specifies the card type as follows:

Column	Content	Meaning
1	C	Comment*
	1	First card of header information set. One such set is required for each file. It must preceed the first environmental simulation set.
	2	First card of environmental simulation set.
	*	Second or continuation card of environmental set.

*All comments prior to header card will be displayed on the instructor's CRT.

Header Information

The header consists of one card whose format is as follows:

Column	Content	Meaning
1	1	Identifies this as header information.
3		Azimuth radar display:
	0	off
	1	on
	2	on, no hashmarks.
5	T,F	Azimuth servo, on if T, else off.
7		Elevation radar display:
	0	off
	1	on
	2	on, no hashmarks.
9	T,F	Elevation servo.
11-12		Minimum number of runs, not used in phase 2.
14-15		Maximum number of runs, not used in phase 2.
17-29		Text file name, text to be displayed on student CRT; or blank.
31-32		PMV number
34-35		PMV number
⋮		
	-1	Indicates the end of freeze PMVs.

Environmental Simulation Card Sets

The environmental simulation is given in two-card sets described below:

Card 1:

Column	Content	Meaning
1	2	First card of an environmental simulation set.
3		Number of error free repeats of this problem.
7		A/C type:
	1	U-21
	2	A6
	3	P3
	4	T38.
9-12	xx.x	Starting range from touchdown in radar miles.
14-17	xx.x	Ending range in miles.
23		Indicate the form of the information in columns 25-33:
	1	A/C starting position specified in relation to environmental simulation, i.e., in feet and miles.
	2	A/C starting position specified in relation to display simulation, i.e., by zone.
25-28		A/C starting altitude in feet or by zone (right justified).
30-33		A/C starting offset in miles (xx.x) negative if left or by zone (right justified).
35		Pilot type: 1 = best, 5 = worst.
37		Type of flight:
	1	Pilot responds normally to controller advisories.
	2	Restrict A/C position to contiguous azimuth zones given in columns 39-43.
	3	Restrict A/C position to contiguous elevation zones given in columns 45-49.

Column	Content	Meaning
	4	Restrict A/C position to contiguous azimuth and elevation zones given in columns 39-49.
39-40		Left azimuth zone (e.g., -2).
42-43		Right azimuth zone (e.g., 1).
45-46		Lower elevation zone (e.g., -2).
48-49		Upper elevation zone (e.g., 1).
51		True if handoff to be given, else false.
53		True if the azimuth target picture is on.
55		True if the elevation target picture is on.

Card 2:

Column	Content	Meaning
1	*	Continuation card.
3		Approach type, used for the handoff message as follows:
	1	Full stop
	2	Low approach
	3	Touch-and-go
	4	Minimum fuel
	5	No-gyro.
5		Clearance information for tower simulation:
	1	Clearance given at first request
	2	Continue, then clear at 2 miles
	3	Not given (no response)
	4	Wave off
	5	Clearance given then cancelled.
		Wind information:
6-9		Mean wind heading
10-12		Mean wind speed in knots
13-15		Mean gust speed in knots
16-20	xx.x	Mean gust duration in seconds
21-24	.xx	Fraction of time gusting occurs
25-29	x.xx	Wind variability (0.00 - 1.00)
30-34	xx.x	Wind speed correlation time
35-38		Ceiling height in feet
40	T,F	If T, pilot has wheels down before controller says "wheels down."
41-45		Number of seconds pilot will wait without radio contact before executing a missed approach.
47	T,F	If T, cause A/C to descend to a point which requires a low altitude alert be given once during the run.
49	T,F	If T, cause a gyro failure after handoff, at the point specified in columns 54-57.
50-54	xx.x	Range from touchdown in miles when gyro failure occurs.

Phase 3 Problem Specification File

Phase 3 problems are scored exercises which allow the student to practice and integrate his new skills. One phase 3 instructon file exists for every task which provides phase 3 training. It provides for both individual problems and randomly selected problems, with scoring of specified performance measurement variables. It is a card image file whose suggested name incorporates both task and phase of instruction, e.g., for task 3.2 the file name should be T03$02.03. The file contains three types of cards: comments, the header information set (two cards) and environmental simulation set (two cards each). The file contains exactly one header information set, and this precedes all environmental simulations ets. In general, the header information pertains to all problems and includes information such as PMVs to be scored, etc. The environmental simulation sets describe a particular problem or set of problems. Note that a sufficient number of exercises is included to provide the maximum number of runs specified. A simple way to ensure this is to specify a multipossibility exercise as the last one in the file.

Card formats are described in tabular form below. The code in column one specifies the card type as follows:

Column	Content	Meaning
1	C	Comment*
	1	First card of header information set. One header information set is required for each phase 3 file; it must preceed the first environmental simulation set.
	2	First card of environmental simulation set.
	*	Second or continuation card of header or environmental set.

*All comments prior to header will be displayed on the instructor's CRT.

Header Information

The header consists of two cards of information. The first is similar to that for phase 2. The second contains the error scores which must not be exceeded in order to pass at this level. These scores are integer percentage error scores which will be compared to the average error score over the last ten problems or over the minimum number of runs, whichever is smaller. The format of the cards is as follows:

Card 1:

Column	Content	Meaning
1	1	Identifies this as header information.
3		Azimuth radar display:
	0	off
	1	on
	2	on, no hashmarks.
5	T,F	Azimuth servo, on if T, else off.
7		Elevation radar display:
	0	off
	1	on
	2	on, no hashmarks.
9	T,F	Elevation servo.
11-12		Minimum number of runs.
14-15		Maximum number of runs.
17-29		Text file name, text to be displayed on student CRT; or blank.
31-32		PMV number relating to this skill (score checked first).
34-35		PMV number relating to this skill.
⋮		⋮
	-1	End of PMV numbers relating to this skill.

Card 2:

Column	Content	Meaning
1	*	Identifies this card as the continuation.
3-5		Maximum allowable % error score to pass PMV1, or -1 if not scored.
7-9		Maximum allowable % error score to pass PMV2, or -1.
⋮		⋮

Environmental Simulation Card Sets

The environmental simulation information is given in two-card sets. These sets are of two types, distinguished by a code in column 3. The first type provides initialization information for one run (which may be repeated). The second type provides a range of information about environmental parameters from which individual problem parameters are chosen randomly.

Single Problem Specification

This card set provides information regarding one run, which may be repeated. Note that if the repeat feature is chosen, no further cards will be examined in this file, rather the system will continue to use this information for problem setup until the conditions for progressing to the next task are met. The format of the cards in this set is shown below. These cards differ from the phase 2 environmental simulation cards only in columns 3 and 5 of the first card.

Card 1:

Column	Content	Meaning
1	2	First card of an environmental simulation set.
3	1	The information is for one run.
5	T,F	T if this same problem is to be repeated, else F.
7		A/C type:
	1	U-21
	2	A6
	3	P3
	4	T38.
9-12	xx.x	Starting range from touchdown in radar miles.
14-17	xx.x	Ending range in miles.
23		Indicates the form of the information in columns 25-33.
	1	A/C starting position specified in relation to environmental simulation, i.e., in feet and miles.
	2	A/C starting position specified in relation to display simulation, i.e., by zone.
25-28		A/C starting altitude in feet or by zone (right justified).
30-33		A/C starting offset in miles (xx.x) negative if left, or by zone (right justified).
35		Pilot type: 1 = best, 5 = worst.
37		Type of flight:
	1	Pilot responds normally to controller advisories.
	2	Restrict A/C position to contiguous azimuth zones given in columns 39-43.
	3	Restrict A/C position to contiguous elevation zones given in columns 45-49.
	4	Restrict A/C position to contiguous azimuth and elevation zones given in columns 39-49.

Card 1 (Cont):

Column	Content	Meaning
39-40		Left azimuth zone (e.g., -2).
42-43		Right azimuth zone (e.g., 1).
45-46		Lower elevation zone (e.g., -2).
48-49		Upper elevation zone (e.g., 1).
51		True if handoff to be given, else false.
53		True if the azimuth target picture is on.
55		True if the elevation target picture is on.

Card 2:

Column	Content	Meaning
1	*	Continuation card.
3		Approach type, used for the handoff message as follows:
	1	Full stop
	2	Low approach
	3	Touch-and-go
	4	Minimum fuel
	5	No-gyro.
5		Clearance information for tower simulation:
	1	Clearance given at first request
	2	Continue, then clear at 2 miles
	3	Not given (no response)
	4	Wave off
	5	Clearance given then cancelled.

Card 2 (Cont):

Column	Content	Meaning
		Wind information:
6-9		Mean wind heading
10-12		Mean wind speed in knots
13-15		Mean gust speed in knots
16-20	xx.x	Mean gust duration in seconds
21-24	.xx	Fraction of time gusting occurs
25-29	x.xx	Wind variability (0.00 - 1.00)
30-34	xx.x	Wind speed correlation time
35-38		Ceiling height in feet
40	T,F	If T, pilot has wheels down before controller says "wheels down."
41-45		Number of seconds pilot will wait without radio contact before executing a missed approach.
47	T,F	If T, cause A/C to descend to a point which requires a low altitude alert be given once during the run.
49	T,F	If T, cause a gyro failure after handoff, at the point specified in columns 54-57.
50-54	xx.x	Range from touchdown in miles when gyro failure occurs.

Multipossibility Problem Specification

This card set provides a range of initial conditions for problem set up. The phase 3 executive selects individual run parameters randomly from those specified, and thus can provide a variety of similiar problems as required for adpative training. Note that, as with the repeat option on a single problem specification, no further cards are examined since the system continues to use this information for problem setup until the conditions for progressing to the next task are met. The format of the cards in this set is shown below.

Card 1:

Column	Content	Meaning
1	2	First card of an environmental simulation set.
3	2	Multi-possibility problem parameters.
		Starting positions. Percentages in columns 5-31 must sum to 100. Handoffs always given.
5-7		% of minimum fuel problems starting in position 1 (right base, minimum fuel, 4 miles, 1000 ft, heading 140° ± variation, clearance always given).*
9-10		Maximum heading variation from 140° on position 1 starts. (|var| ≤ 10°).
12-14		% of problems starting in position 2 (right base, 8 miles, 1500 ft, heading 140° ± variation).*
19-21		% of problems starting in position 3 (straight in approach, 11 miles, 1500 ft. heading 160° ± variation).*
23-24		Maximum heading variation on position 3 starts.
26-28		% of problems starting in position 4 (left base, 8 miles, 1500 ft, heading 180° ± variation.)*

*Note: These are track headings. The simulated pattern controller will supply appropriate crab to the assigned heading to produce the specified track heading.

Card 1 (Cont):

Column	Content	Meaning
30-31		Maximum heading variation on position 4 starts.
		Aircraft speeds. Sum of values in columns 33-43 must equal 100.
33-35		% slow A/C.
37-39		% medium A/C.
41-43		% fast A/C.
		Pilot type. Sum of values in columns 45-63 must be 100.
45-47		% type 1 (best).
49-51		% type 2.
53-55		% type 3.
57-59		% type 4.
61-63		% type 5 (worst).

Card 2:

Column	Content	Meaning
1	*	Continuation card.
		Approach type. Sum of values in columns 3-13 must be 100. Applies only for starting positions 2-4. Minimum fuel is full stop.
3-5		% full stop.
7-9		% low approach
11-13		% touch-and-go.

Card 2 (Cont):

Column	Content	Meaning
15-17		% no-gyro approaches. These gyro failures will be uniformly distributed between 3 and 5 miles. If the failure occurs at 5+ miles (or 3+ miles on minimum fuel approaches), it will be announced by the pattern controller in the handoff message.
		Clearance information for tower simulation. Note: Clearance will always be given on minimum fuel approaches. These values apply only to other types of approaches. Sum of values in columns 23-41 must be 100.
19-21		% clearance given at first request.
23-25		% continue, then clear at 2 miles.
27-29		% not given (no response).
31-33		% waveoff
35-37		% clearance given then cancelled.
39-41		Number of seconds pilot waits without radio contact before executing a missed approach.
		Wheels down. Sum of columns 43-49 must equal 100.
43-45		% wheels down.
47-49		% wheels not down.
		Ceiling visibility. Sum of columns 51-57 must equal 100.
51-53		% runway visible at decision height.
55-57		% runway not visible at decision height.

Card 3:

Column	Content	Meaning
1	*	Continuation card.
		Wind variability. Sum of columns 3-21 must equal 100.
3-5		% wind variability parameter .1, wind speed correlation time 30 seconds, and gusting occurring 5% of the time.
7-9		% wind variability parameter .25, correlation time 15 seconds, and gusting 10% of the time.
11-13		% wind variability parameter .5, correlation time 15 seconds, and gusting 20% of the time.
15-17		% wind variability parameter .75, correlation time 12.5 seconds, and gusting 30% of the time.
19-21		% wind variability parameter 1., correlation time 10 seconds, and gusting 40% of the time.
		Wind gustiness. Sum of columns 23-41 must equal 100.
23-35		% gust speed 5 knots, duration 3 seconds.
27-29		% gust speed 5 knots, duration 10 seconds.
31-33		% gust speed 10 knots, duration 10 seconds.
35-37		% gust speed 15 knots, duration 10 seconds.
39-41		% gust speed 20 knots, duration 10 seconds.
		Wind speed. Sum of columns 43-61 must equal 100.

Card 3 (Cont):

Column	Content	Meaning
43-45		% mean wind speed 0 knots.
47-49		% mean wind speed 5 knots.
51-53		% mean wind speed 10 knots.
55-57		% mean wind speed 15 knots.
59-61		% mean wind speed 30 knots.
		Wind direction. The following directions are the absolute value of the direction, relative to the runway heading (160°). The sign of the wind direction is chosen randomly at the beginning of each run. Sum of columns 63-80 must equal 100.
63-65		% 0°
67-69		% 15°
71-73		% 30°
75-77		% 60°
79-80		% 90°

Performance Run Specification File

The performance run (P-run) will be used to determine whether or not the student passes the course. Because of its importance, final scoring will be deferred until the trainee and the instructor have the opportunity to review the replay of the run and correct any misrecognitions. The file consists of three types of cards: comments, the header card, and an environmental simulation card set, distinguished by a code in column 1 as follows:

Column	Content	Meaning
1	C	Comment*
	1	Header information.
	2	First card of environmental simulation set.
	*	Second or continuation card of environmental set.

*Comments prior to the header card will be displayed on the instructor's CRT.

Header Information

The header consists of two cards whose format is identical to the phase 3 header.

Environmental Simulation Card Set

One environmental simulation card set must be provided to described the P-run. The set differs from the phase 2 environmental simulation cards only in columns 3 and 5 of the first card.

Column	Content	Meaning
1	2	First card of an environment simulation set.
7		A/C type:
	1	U-21
	2	A6
	3	P3
	4	T38.
9-12	xx.x	Starting range from touchdown in radar miles.
14-17	xx.x	Ending range in miles.

Column	Content	Meaning
23		Indicates the form of the information in columns 25-33.
	1	A/C starting position specified in relation to environmental simulation, i.e., in feet and miles.
	2	A/C starting position specified in relation to display simulation, i.e., by zone.
25-28		A/C starting altitude in feet or by zone (right justified).
30-33		A/C starting offset in miles (xx.x) negative if left or by zone (right justified).
35		Pilot type: 1 = best, 5 = worst.
37		Type of flight:
	1	Pilot responds normally to controller advisories.
	2	Restrict A/C position to contiguous azimuth zones given in columns 39-43.
	3	Restrict A/C position to contiguous elevation zones given in columns 45-49.
	4	Restrict A/C position to contiguous azimuth and elevation zones given in columns 39-49.
39-40		Left azimuth zone (e.g., -2).
42-43		Right azimuth zone (e.g., 1).
45-46		Lower elevation zone (e.g., -2).
48-49		Upper elevation zone (e.g., 1).
51		True if handoff to be given, else false.

Card 2:

Column	Content	Meaning
1	*	Continuation card.
3		Approach type, used for the handoff message as follows:
	1	Full stop
	2	Low approach
	3	Touch-and-go
	4	Minimum fuel
	5	No-gyro.
5		Clearance information for tower simulation:
	1	Clearance given at first request
	2	Continue, then clear at 2 miles
	3	Not given (no response)
	4	Wave off
	5	Clearance given then cancelled.
		Wind information:
6-9		Mean wind heading
10-12		Mean wind speed in knots
13-15		Mean gust speed in knots
16-20	xx.x	Mean gust duration in seconds
21-24	.xx	Fraction of time gusting occurs
25-29	x.xx	Wind variability (0.00 - 1.00)
30-34	xx.x	Wind speed correlation time
35-38		Ceiling height in feet
40	T,F	If T, pilot has wheels down before controller says "wheels down."
41-45		Number of seconds pilot will wait without radio contact before executing a missed approach.
47	T,F	If T, cause A/C to descend to a point which requires a low altitude alert be given once during the run.
49	T,F	If T, cause a gyro failure after handoff, at the point specified in columns 54-57.
50-54	xx.x	Range from touchdown in miles when gyro failure occurs.

Demonstration Run Specification File (DEMOPROBS)

The demonstration runs that take place while the system is idle are chosen randomly from the set of possibilities provided in this file. The file consists of a two card set which is identical in format to the Phase 3 multipossibility environmental card set described previously.

Remedial Training File

Remedial training problems (other than rule explanations for knowledge items) are selected from this file. The file is a randomly organized ASCII text file with 8 word records.

Record	Word	Content	Meaning
1	1	Δn	Where 'n' is the phase of the remedial task for PV01.
		or -1	Where '-1' indicates that there is no remedial problem for PV01.
	2-8		The name of the remedial task (Tnn$mn.pp)
2	1	Δn	
		or -1	
⋮	2-8		
19	⋮		

(one for each PV)

Text File

A specially formatted text file is associated with each phase 1 instruction file and with each problem specification file. The association with phase 1 instruction files is maintained through file names as follows: the instruction file is assigned a unique name beginning with the letter "T." The corresponding text file name begins with "TX" and is otherwise the same as that of the instruction file. Thus for example the instruction file T01$20.01 has associated with it the text file TX01$20.01. Although this naming convention is maintained for problem specification and text file pairs, the association is explicitly specified in the problem files and there is no necessary relation between the names.

The text files are 80 bytes per record binary files whose format is shown below.

Record	Word	Content
1-8	1	Page 1 data: starting record
	2	line count
	3	Page 2 data: starting record
	4	line count
	.	.
	.	.
	.	.
9	1-40	Page 1, first line of text
10	1-40	Page 1, second line of text
.	.	.
.	.	.
.	.	.

Error Explanation File (ERXFI)

This is a file composed of text and binary information used to output error explanations. It has a record length of 43 words. The first two words of each record are pointers used to output 'state-of-the-world' information and are explained below. These two numbers are processed sequentially to provide feedback to the trainee. The remaining 40 words of each record are error messages stored as ASCII text with a carriage return at the end. There are three possible error messages for each error the GCA-CTS detects. When possible, the first two provide descriptions of the error worded in slightly different ways. The third gives the rule or a statement about the consequences of the error and is randomly appended to the first two. The actual text of the error messages is given in Appendix M.

Contents of Words 1 and 2 of Each Record	Information Output
-2	Always print the third line of the explanation
-1	Provide no additional feedback for this error
0	"You were understood to say: [most recent SUS phrase]"
1	"The correct frequency is: [correct frequency]"
2	"The correct call sign is: [correct call sign]"
3	Reserved for correct range for "approaching glidepath"
4	Reserved for present range
5	Reserved for previous turn
6	"The correct azimuth position call is: [correct azimuth position call]"
7	"The correct glidepath position call is: [correct glidepath position call]"
8	"The correct trend message is: [correct trend call]"
9	Reserved for highest priority call at decision height (actually handled by bits in the PMV)
10	"The correct wind information is: [correct wind information]"
11	Reserved for proper missed approach procedure
12	"The reference point is: [issuance of waveoff/ decision height/landing threshold]"
13	"The correct button is: [correct button]"
14	Reserved for no-gyro information
15	"The type of approach is: [type of approach]"

Error Explanation Index File (ERBLK)

This is a binary file of indexes into ERXFI. It has the default record length. It is expected that the file will be read into a two-dimensional array in a statement something like:

```
READ BINARY (NCERIN) ((ERINDEX (BIT, WORD), WORD = 1,19), BIT = 0,15)
```

Subsequently ERXFI can be read with a statement such as:

```
CALL READR (NCERX, ERINDEX (BIT, WORD), BUFF, 1, IER)
```

to get an explanation record corresponding to the bit of a PV word.

Votrax Phrase File (FRAZ.VO)

FRAZ.VO is a contiguous file of octal phoneme codes which the Votrax uses to produce audible output. Each record consists of 64 octal words padded with "-1". The file is indexed by the mnemonic record numbers defined in VX.CO. The high order two bits of each octal word contain information on inflection and the bottom six contain phonemes. One GCA phrase is represented by each record. The contiguous octal format ensures that speed of execution will be optimal.

Votrax Text File (VOTEXT)

This file contains lower case text for the SUS and VOTRAX phrases. The record size is 40 words. Each record contains the text equivalent of a SUS phrase with zeros packed to the right. Each SUS phrase number (as defined in VX.CO) is the record number for its text in VOTEXT. That is, to get the text for '1 mile' one can use a statement like:

```
CALL READR (NCPH, VX1MI, BUFF, 1, IER)
```

Since the records are packed with zeros, more than one phase can be printed per line. This allows one to print full GCA-CTS messages on one line.

Trainee Independent Speech Data File (SPK.VO)

SPK.VO contains trainee independent speech data. These include phrases, identification tags, phrase VRP record pointers, phrase IFP record pointers, number of IFPs necessary for VRP formation, and student CRT phrase prompts. They are arranged in the 32 word per record file as follows:

Record Number	Content
1-4	ID tags
5-8	VRP pointers (for VRP.VO)
9-12	IFP pointers (for IFP.VO)
13-16	IFP minimum
17+	Prompts, 1 record/phrase

Voice Input Feature Patterns (IFP.VO)

IFP.VO is composed of 32 word records. Records 1 through 4 contain the number of IFPs available for VRP formulation. This number may range from 0 (no IFP has been collected for the phrase) to 4 or 10 (maximum number of IFPs have been collected for the phrase). Records 5 through 8 hold pointers to the next IFP storage slot. The pointer references an empty storage slot or the oldest IFP in storage. The remaining records are allocated for actual IFP storage. Each phrase is represented by a maximum of 4 or 10 of the most recently collected IFPs. Distinct phrases require 4 IFPs for VRP formulation whereas less distinguishable phrases require 10 IFPs. All phrase IFPs and their references are stored in phrase identification order. This file resides within the trainee's own subdirectory on the removable disk.

Voice Reference Patterns (VRP.VO)

VRP.VO is also arranged in phrase identification order and 32 word-records. VRP present flags comprise records 1 through 4 of VRP.VO. These flags are set upon phrase VRP formation. Records 5 through 8 store validation percentages for the VRPs which have been validated. Remaining records are given to VRP storage. The VRPs, like the IFPs are stored in 32 time slot (64-word) or 16 time slot (32-word) format as specified by syllabic length (phrases of three syllables or less are represented by 16 time slot VRPs). Filler blocks of one record length are utilized to maintain VRP starts at 1024-word boundaries for window mapping purposes. This file also resides in the trainee's subdirectory.

Special Purpose Digitized Speech Files

CANFILE contains a selection of prerecorded phrases which will be played back upon request. Each record is 1024 words long.

ICANFILE is organized the same as the student's IDVFILE and accesses the phrases stored in CANFILE. Up to 27 phrases may be accessed. At run time these addresses are loaded into a 2 by 27 array as opposed to accessing the particular file record when addresses are desired, the method used with IDVFILE. Each record is two words long; one word for the starting address, one for the ending address.

Prompting Speech Data Files

SCANFILE contains the recorded digitized speech of the student from phase 1. Once recorded it is used as a prompt to elicit the student's vocal responses. Each record is 1024 words long.

ICANFILE is made up of two word index records containing the starting and ending address of each of the phrases recorded in SCANFILE. The record size for this index file is two words, one for the starting address of the phrase and one for the ending address. Each record corresponds to a phrase and progresses sequentially, such that record one references phrase one.

Speech Data Replay Files (RPLSPH and RPPSPH)

RPLSPH contains the uninterrupted recording of an entire phase 3 run. Upon replay this file is synchronously played back to simulate the student's voice in the original run. RPPSPH is the equivalent file for performance runs. Each record is 1024 words long.

IDVFILE is a simple two word file containing the starting and ending address of RPLSPH. PIDVFILE is the equivalent file for the performance run.

Radar Data Replay File (RPLDSP and RPPDSP)

These files are written by RADAR and read during replay to recreate the radar display. RRLDSP is written during phase 3 runs and RPPDSP is written during a P-run. These are random files, with eight words per record. Their contents are described below.

Word	Contents
1	X_1, coordinate of target. -32000 if PCMSG should = 2.
2	Y_1, lower coordinate of azimuth target.
3	Y_2, upper coordinate of azimuth target.
4	Y_3, lower coordinate of elevation target.
5	Y_4, upper coordinate of elevation target.
6	Y_A, azimuth servo position.
7	Y_E, elevation servo position.
8	X_2, wind speed and heading bits 1-7 and 8-16, respectively.

Activity Replay File (RPLACT and RPPACT)

Display starting conditions, all recognition information, student panel inputs, servo inputs and synthesized speech outputs are stored in these files. RPLACT is written during a phase 3 run, RPPACT is written during the P-run. They are randomly organized, with eight words per record. There are six types of records which are distinguished by a code in column 1. These records are described below.

Record Type	Word	Content	Meaning
End of file	1	-1	Identifies this as the last record of the replay file.
	2-8		Unused
Header	1	1	Identifies this as header record.
	2		CTHTBL (0), ideal final course heading with crab.
	3	T,F	SASTRIN, true if straight in approach.
	4	T,F	T if picture 1 is on PCAZIM.
	5	T,F	T if picture 2 is on PCAZHASH.
	6	T,F	T if picture 3 is on PZELEV.
	7	T,F	T if picture 4 is on PCELHASH.
	8	T,F	T if picture 7 is on PCSWEEP.
Header, record 2	1	T,F	True if picture 8 is on PCAZTARG.
	2	T,F	True if picture 9 is on PCAZTRAIL.
	3	T,F	True if picture 10 is on PCELTARG.
	4	T,F	True if picture 11 is on PCELTRAIL.
	5		Initial azimuth servo Y coordinate.
	6		Initial elevation servo Y coordinate.
	7		CTISIGNX, right or left base indicator.
	8	T,F	True if feeder gives handoff.

Record Type	Word	Content	Meaning
Header, record 3	1	PTAPR	Approach type
	2	CTFREQ	Button frequency for run
	3	ACTYP	Aircraft type
	4	CTREL	T if pattern controller to release frequency without being asked
	5	PTWEEL	T if wheels down
	6	CTCLR	Clearance type
	7	EMGYFL	T if gyro failure
	8	CTHOF	T if handoff to be given
Replay Synchronization	1	2	Identifies this as a replay synchronization record.
	2		Time in .5 second ticks from the start of the problem.
	3		Digitized speech record number.
	4-8		Unused
SUS Output	1	3	Identifies this as a recognition block.
	2		Time of LP_4 in .5 second ticks from the start of the problem.
	3		Time in 100 msec ticks from above timer.
	4		First choice message understood (message actually given to APE).
	5		Heading, if any, or -2 if digits not recognized, -1 if no heading.
L=Left 3 bits	6L		Call sign or all bits set.
R=Right byte	6R		Wind speed or all bits set.

Record Type	Word	Content	Meaning
	7		Second choice message understood, or -1.
	8	T,F	T if correction was applied to this phrase.
A=Left byte	9A	CTGPP	Correct glidepath position message.
	9B	CTGPT	Correct glidepath trend message.
B=Right byte	10A	CTCRP	Correct course position message.
	10B	CTCRT	Correct course trend message.
	11A	CTOTHR	Other than range/emergency message.
	11B	1,0	1, if KYMIKE=true, 0, if false.
	12A	0-12	Azimuth zone + 6.
	12B	0-12	Elevation zone + 6.
	13	± 1	Azimuth trend sign
	14	± 1	Elevation trend sign.
	15		Miles to touchdown x 100.
	16		Aircraft magnetic heading.
Panel Changes	1	4	Identifies this as a panel change.
	2		Time in .5 sec ticks.
	3		DIA word.
	4		DIB word.
	5		DOA word.
	6		DOB word.
	7		Miles from touchdown x 100.
	8		Not used.

Record Type	Word	Content	Meaning
Automated Voice Output	1	6	Identifies this as GLIB output information.
	2		Time in .5 sec ticks.
	3		Miles to touchdown X 100.
	4		First phrase to be output.
	⋮		
	n	-1	End of phrases to be output.
	n+1-8		Not used.
Special scoring timers, etc.	1	7	Identifies this as a special scoring parameter record
	2	1	Identifies this as an event detected record
	3	1-12	1: 50% of target appears 2: "C/S, Radar Contact" given 3: A/C entering zone 3 from zone 2 4: Low altitude condition exists 5: Pattern controller has given handoff to final controller 6: Pilot has begun executing waveoff 7: Targets are within 2.8 miles of one another 8: Target has transited between "well" zones in 3 seconds or less 9: Target has moved from one glide-path zone to another 10: Radar contact lost 11: "Approaching glidepath" end of window 12: "Approaching glidepath" beginning of window
	4		Time in half-second ticks
	5-8		Unused
	1	7	As above
	2	2	Identifies this as a mile record
	3		Miles to touchdown x 100

Record Type	Word	Content	Meaning
	4		Azimuth display coordinate for aircraft center
	5		Azimuth zone
	6		Time in half-second ticks
	7-8		Unused
	1	7	As above
	2	3	Identifies this as a critical range record
	3	1-3	1: ADH 2: OLT 3: other ranges of interest used to trigger RNGCAL
	4		Miles to touchdown x 100
	5		Time in half-second ticks
	6-8		Unused
	1	7	As above
	2	4	Identifies this as a wind record
	3		Wind heading for clearance
	4		Wind heading for clearance
	5-8		Unused
	1	7	As above
	2	5	Identifies this as a no-gyro record
	3		Azimuth trend
	4		Azimuth zone
	5		Miles to touchdown x 100
	6		Time in half-second ticks
	7-8		Unused

Student Error File (ER and PER)

Time in LP4 clock ticks of error, record number of error explanation in ERXFI and the number of the PMV with the error are in this file. ER is written during phase 3 runs and PER is written during a P-run. It is randomly organized and has a record length of four words.

Word	Meaning
1	Time of error in LP4 clock ticks
2	Record number in ERXFI of error explanation
3	Number of PMV with the error
4	Unused (0)
⋮	
1 (of last record)	End of file marker (32000)

Student Training Task Activity File (SR1)

24 Word Record

Record #	
1	Student's name (24 words)
2	Student's ID number (24 words)
3	Unused
4-11	Task summary blocks (24 words)
⋮	
N	

Task Summary Blocks:

Word #			
1	Adjusted Phase of task	(1-4) (7-9) (10-12)	1-PRUN 1-3 (override) 1-3 (remedial)
2-8	Name of task		
9-11	Date started		
12-14	Time started		
15	Pointer to beginning of phase 3 problem blocks		
16	Pointer to last phase 3 problem block or -1		
17	Pointer to summary block or -1		
18	Final disposition -1 = not yet done, 0 = passed, 1 = overridden, 2 = continued by instructor, 3 = challenged, 4 = phase 3, not passed, 5 = phase 3 not passed (too few problems in task file)		
19	Number of tries or -1		
20	Number of time-outs that occurred during task or -1		

AD-A087 190 LOGICON INC SAN DIEGO CA TACTICAL AND TRAINING SYSTE--ETC F/G 17/9
GROUND CONTROLLED APPROACH CONTROLLER TRAINING SYSTEM (GCA-CTS)--ETC(U)
JUN 80 G BARBER, J BOLLENBACHER, D BREWTON N61339-77-C-0162

UNCLASSIFIED NAVTRAEQUIPC-77-C-0162-3 NL

8 OF 8
AD
A087190

END
DATE
FILMED
9-80
DTIC

Task Summary Blocks (Cont):

Word #

21-24 Unused, but = -1

NOTE: Words 16-24 are not known until the task is completed. Between the start of the task and its completion, their value is -1.

Words 15, 16 and 17 are used only for phase 3 type tasks.

Pointers in words 15-17 are in record numbers.
15 and 16 point to student file: P3.
17 points to student file: SUM.

Sign On, Sign Off and Alignment Activity File (PV19)

Description: This file contains Sign on and Sign off times for the student and the alignment scoring data.

Block size: 24 bytes.

Format: Binary data.

Contents of each Record:

Word #	Contents
1-3	Date of sign on
4-6	Time of sign on
7	PVE(9) or -1, if alignment was not scored
9-11	Time of sign off
12	Not used

Phase 3 Task Summary File (SUM)

Description: This file contains average PVE scores for each phase 3 task.

Block size: 64 bytes.

Format: Binary data.

Contents of each record:

Word #	Contents
1-7	FNPHZ (name of task)
8-10	Date of scoring
11-13	Time of scoring
14-31	Average PVE score
32	Not used

Phase 3 Problem Description File (P3)

Description: Contains the error scores and performance data for individual phase 3 problems.

Block size: 240 bytes.

Format: Binary data.

Contents of each record:

Word #	Contents
1-3	Date of scoring
4-6	Time of scoring
7	GZADAPTPT (true if pilot type was adapted)
8	GZADAPTAC (true if aircraft type was adapted)
9	GZADAPTWN (true if wind was adapted)
10-29	PVE (scores for this approach)
30-99	Dump of PMVC.CO from PV00 through PV19
100	ACTYP (aircraft type)
101	CTREL (true if pattern controller releases frequency on "radar contact")
102	CTCLR (clearance type)
103	EMGYFL (true if gyro failure is to occur)
104	PTYP (pilot type)
105	PTAPR (type of approach)
106,7	ENWHT (wind parameters)
108,9	ENMWS (wind parameters)
110,111	ENMGS (wind parameters)
112,113	ENMAGS (wind parameters)
114	ENSCAT (starting position)
115	ENGCAT (PTWEEL - indicates pilot's response to radio check)
116,117	ENCEIL (ceiling)
118	GZMNR (minimum number of problems)
119	GZNR (maximum number of problems)
120	-1

Student Status File (SCRATCH)

Description: The scratch file contains temporary data about an individual student's position in the course and pending special requests.

Block size: 48 bytes.

Format: Binary data.

Contents of each record #1:

Word #	Initial Value	Contents
1	0	GZPHZ
2	0	GZRUN
3	false	GZREM
4-10	0	FNPHZ
11	4	TZSR1
12	1	TZP3
13	1	TZSUM
14-15	0	TZCSYL
16-17	0	GZCSYL
18	0	GZSEED
19	true	GZDONE
20	0	TZLSUM
21	false	GZSC19
22	0	GZT17
23	1	TZPV19
24	-1	Not used

Contents of each record #2:

Word #	Initial Value	Contents
1	0	GZPTRY
2	0	GZTRY
3	-1	GZOR
4	0	GZOPHZ
5-11	0	GTZISR
12	0	GZTOT
13	false	KBVSTRT
14-15	0	TZPHZ
16	false	GZSDSPTAC
17	false	GZSOPTNF
18	false	GZADAPTPT
19	false	TZPV19
20-24	false	Not used

Contents of Record #3:

Word #	Initial Value	Contents
1-24	false	GZPILL

Contents of Record #4:

Word	Initial Value	Contents
1-24	0	name of remedial training file (ASCII)

Voice Test Activity File (LOG.VT)

This file contains information about the student's use of the voice test feature of the GCA-CTS. Each time the student enters or exits the voice test mode (except when directed to do so by a phase 1 instruction) an eight word record is written. The format of these records is:

Word #	Contents
1-3	Date
4-6	Time
7	0 for INIT VOICE TEST, 1 for STOP VOICE TEST
8	T,F (true if instructor functions are active at the student station)

New R/T Activity File (LOG.RT)

This file contains information about the student's use of the New R/T feature of GCA-CTS. Each time the student enters or exits the New R/T mode a record is kept showing when retraining began and which phrases were retrained. These records have a variable length depending on how many phrases were trained. The format is as follows:

Word #	Contents	Meaning
1	0	Marks the beginning of a new record.
2-4	Date	
5-7	Time	
8	T,F	True if instructor functions were available at the student station.
9	Phrase #	First phrase retrained.
10	Phrase #	Second phrase retrained.
	⋮	
n	Phrase #	Last phrase retrained.

APPENDIX D

COMPILE MACROS

```
/CP1.MC
/MACRO TO COMPILE CPU1 ROUTINES

DELETE ERROR
FORTRAN/S/B/I   $LPT/L ERROR/E ACDMP.FR
FORTRAN/S/B/I   $LPT/L ERROR/E ACSET.FR
FORTRAN/S/B/I   $LPT/L ERROR/E ACTIVITY.FR
MAC             $LPT/L ERROR/E ACTOUT.SR
FORTRAN/S/B/I   $LPT/L ERROR/E ACTSUS.FR
FORTRAN/S/B/I   $LPT/L ERROR/E ACVERT.FR
FORTRAN/S/B/I/X   $LPT/L ERROR/E ADAPT.FR
FORTRAN/S/B/I   $LPT/L ERROR/E AFAPOP.FR
FORTRAN/S/B/I   $LPT/L ERROR/E AFDNA.FR
FORTRAN/S/B/I   $LPT/L ERROR/E AFWC.FR
FORTRAN/S/B/I   $LPT/L ERROR/E APENIT.FR
FORTRAN/S/B/I   $LPT/L ERROR/E APE1NIT.FR
FORTRAN/S/B/I   $LPT/L ERROR/E APE2NIT.FR
FORTRAN/S/B/I   $LPT/L ERROR/E APE3NIT.FR
FORTRAN/S/B/I   $LPT/L ERROR/E APE4NIT.FR
FORTRAN/S/B/I   $LPT/L ERROR/E APE5NIT.FR
FORTRAN/S/B/I   $LPT/L ERROR/E APEX.FR
FORTRAN/S/B/I   $LPT/L ERROR/E APOP.FR
FORTRAN/S/B/I   $LPT/L ERROR/E APRAX.FR
FORTRAN/S/B/I   $LPT/L ERROR/E APREX.FR
FORTRAN/S/B/I   $LPT/L ERROR/E ATRPLY.FR
FORTRAN/S/B/I   $LPT/L ERROR/E BEATIT.FR
FORTRAN/S/B/I   $LPT/L ERROR/E BEODES.FR
FORTRAN/S/B/I   $LPT/L ERROR/E BLOCK1.FR
FORTRAN/S/B/I   $LPT/L ERROR/E BUTX.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CK120.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKACK.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKADH.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKAGP.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKBD.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKCHK.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKCLR.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKCN.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKCOR.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKCRP.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKCWO.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKEZN.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKFCP.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKGMR.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKGPP.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKHDCOR.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKHN.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKHO.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKICS.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKIN.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKK3.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKK5.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKLAA.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKNOA.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKOLT.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKOVR.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKP18.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKPAT.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKPCLR.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKRFR.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKRNG.FR
```

```
FORTRAN/S/B/I   $LPT/L ERROR/E CKROM.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKTB.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKTLS.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKWO.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CKZN3.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CLEAR.FR
MAC             $LPT/L ERROR/E CLOK.SR
FORTRAN/S/B/I   $LPT/L ERROR/E CLRBUTX.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CLREG.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CLRNC.FR
FORTRAN/S/B/I   $LPT/L ERROR/E COMBO.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CONCEIVETH.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CONTOW.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CRSTUFE.FR
FORTRAN/S/B/I   $LPT/L ERROR/E CSOVER.FR
FORTRAN/S/B/I   $LPT/L ERROR/E DECK.FR
FORTRAN/S/B/I   $LPT/L ERROR/E DEDUCETHEC.FR
FORTRAN/S/B/I   $LPT/L ERROR/E DEMO.FR
FORTRAN/S/B/I   $LPT/L ERROR/E DESCRPROB.FR
FORTRAN/S/B/I   $LPT/L ERROR/E DESEL.FR
FORTRAN/S/B/I   $LPT/L ERROR/E DHCK.FR
FORTRAN/S/B/I   $LPT/L ERROR/E DIE.FR
FORTRAN/S/B/I   $LPT/L ERROR/E DIGIN.FR
FORTRAN/S/B/I   $LPT/L ERROR/E DIRT.FR
MAC             $LPT/L ERROR/E DISPATCH.SR
FORTRAN/S/B/I   $LPT/L ERROR/E DONE.FR
MAC             $LPT/L ERROR/E PARTITION/S DPART.SR DPART.RB/6
FORTRAN/S/B/I   $LPT/L ERROR/E DWAIT.FR
FORTRAN/S/B/I   $LPT/L ERROR/E ENDAPOP.FR
FORTRAN/S/B/I   $LPT/L ERROR/E ENDFEED.FR
FORTRAN/S/B/I   $LPT/L ERROR/E ERIN.FR
FORTRAN/S/B/I   $LPT/L ERROR/E ERLOOKUP.FR
FORTRAN/S/B/I   $LPT/L ERROR/E ERRHAN.FR
FORTRAN/S/B/I   $LPT/L ERROR/E ERRTEST.FR
FORTRAN/S/B/I   $LPT/L ERROR/E EX1PERT.FR
MAC             $LPT/L ERROR/E EXEC.SR
FORTRAN/S/B/I   $LPT/L ERROR/E EXPERT.FR
FORTRAN/S/B/I   $LPT/L ERROR/E EXPLAIN.FR
FORTRAN/S/B/I   $LPT/L ERROR/E FIACINIT.FR
FORTRAN/S/B/I   $LPT/L ERROR/E FB19.FR
FORTRAN/S/B/I   $LPT/L ERROR/E FEED.FR
FORTRAN/S/B/I   $LPT/L ERROR/E FILL.FR
FORTRAN/S/B/I   $LPT/L ERROR/E FILNM.FR
FORTRAN/S/B/I   $LPT/L ERROR/E FINCON.FR
FORTRAN/S/B/I   $LPT/L ERROR/E FOR1.FR
FORTRAN/S/B/I   $LPT/L ERROR/E FOR2.FR
FORTRAN/S/B/I   $LPT/L ERROR/E FOR3.FR
FORTRAN/S/B/I   $LPT/L ERROR/E FOR4.FR
FORTRAN/S/B/I   $LPT/L ERROR/E FR301.FR
FORTRAN/S/B/I   $LPT/L ERROR/E FR304.FR
FORTRAN/S/B/I   $LPT/L ERROR/E FR3HELP.FR
FORTRAN/S/B/I   $LPT/L ERROR/E FR912.FR
FORTRAN/S/B/I   $LPT/L ERROR/E FRDIALOG.FR
FORTRAN/S/B/I   $LPT/L ERROR/E FRREST.FR
FORTRAN/S/B/I   $LPT/L ERROR/E FTHSET.FR
FORTRAN/S/B/I   $LPT/L ERROR/E GAMOD.FR
FORTRAN/S/B/I   $LPT/L ERROR/E GETANS.FR
FORTRAN/S/B/I   $LPT/L ERROR/E GETBUFF.FR
FORTRAN/S/B/I   $LPT/L ERROR/E GETDIR.FR
FORTRAN/S/B/I   $LPT/L ERROR/E GETNEXT.FR
FORTRAN/S/B/I   $LPT/L ERROR/E GIMMIE.FR
FORTRAN/S/B/I   $LPT/L ERROR/E GLBF.FR
MAC             $LPT/L ERROR/E GLIB.SR
FORTRAN/S/B/I   $LPT/L ERROR/E GO.FR
FORTRAN/S/B/I   $LPT/L ERROR/E GOOF1.FR
FORTRAN/S/B/I   $LPT/L ERROR/E GPRUN.FR
```

```
FORTRAN/S/B/I   $LPT/L ERROR/E GREAL.FR
FORTRAN/S/B/I   $LPT/L ERROR/E GRESP.FR
FORTRAN/S/B/I   $LPT/L ERROR/E GTREND.FR
FORTRAN/S/B/I   $LPT/L ERROR/E GYROKILL.FR
FORTRAN/S/B/I   $LPT/L ERROR/E HEAD2.FR
FORTRAN/S/B/I   $LPT/L ERROR/E HEAD3.FR
FORTRAN/S/B/I   $LPT/L ERROR/E HED4.FR
FORTRAN/S/B/I   $LPT/L ERROR/E HEYFEED.FR
FORTRAN/S/B/I   $LPT/L ERROR/E HEYTZEC.FR
FORTRAN/S/B/I   $LPT/L ERROR/E HOCK.FR
FORTRAN/S/B/I   $LPT/L ERROR/E HOLD.FR
FORTRAN/S/B/I   $LPT/L ERROR/E HOSAY.FR
FORTRAN/S/B/I   $LPT/L ERROR/E HOWFAR.FR
FORTRAN/S/B/I   $LPT/L ERROR/E HOWHIGH.FR
FORTRAN/S/B/I   $LPT/L ERROR/E HOWNOW.FR
MAC             $LPT/L ERROR/E IADR.SR
FORTRAN/S/B/I   $LPT/L ERROR/E IGNORE.FR
FORTRAN/S/B/I   $LPT/L ERROR/E IGOODKY.FR
FORTRAN/S/B/I   $LPT/L ERROR/E IKBRD.FR
FORTRAN/S/B/I   $LPT/L ERROR/E IMOFF.FR
FORTRAN/S/B/I   $LPT/L ERROR/E INITRT.FR
FORTRAN/S/B/I   $LPT/L ERROR/E IPBIN1.FR
MAC             $LPT/L ERROR/E IPBOUT1.SR
FORTRAN/S/B/I   $LPT/L ERROR/E ISABUF.FR
FORTRAN/S/B/I   $LPT/L ERROR/E ISAY.FR
FORTRAN/S/B/I   $LPT/L ERROR/E IVT.FR
FORTRAN/S/B/I   $LPT/L ERROR/E KPROC.FR
FORTRAN/S/B/I   $LPT/L ERROR/E KREPLAY.FR
FORTRAN/S/B/I   $LPT/L ERROR/E KSTUD.FR
FORTRAN/S/B/I   $LPT/L ERROR/E KTEACH.FR
FORTRAN/S/B/I   $LPT/L ERROR/E LEVEL1.FR
FORTRAN/S/B/I   $LPT/L ERROR/E LIST.FR
MAC             $LPT/L ERROR/E LOCOSYM/S LOCO1.SR CRAZY1.SR
MAC             $LPT/L ERROR/E LOCOSYM/S LOCO2.SR CRAZY2.SR
MAC             $LPT/L ERROR/E LOCOSYM/S LOCO3.SR CRAZY3.SR
MAC             $LPT/L ERROR/E LOOKATME.SR
FORTRAN/S/B/I   $LPT/L ERROR/E LOOKUP.FR
FORTRAN/S/B/I   $LPT/L ERROR/E LOST.FR
FORTRAN/S/B/I   $LPT/L ERROR/E LOW.FR
FORTRAN/S/B/I   $LPT/L ERROR/E MARKIT.FR
FORTRAN/S/B/I   $LPT/L ERROR/E MENU.FR
FORTRAN/S/B/I   $LPT/L ERROR/E MILER.FR
FORTRAN/S/B/I   $LPT/L ERROR/E MODELINIT.FR
FORTRAN/S/B/I   $LPT/L ERROR/E MODIFY.FR
FORTRAN/S/B/I   $LPT/L ERROR/E MODWIND.FR
FORTRAN/S/B/I   $LPT/L ERROR/E MOVEPILOT.FR
MAC             $LPT/L ERROR/E MOVIT.SR
FORTRAN/S/B/I   $LPT/L ERROR/E MSGFILL.FR
FORTRAN/S/B/I   $LPT/L ERROR/E MSGPICKED.FR
FORTRAN/S/B/I   $LPT/L ERROR/E NEWADVISOR.FR
FORTRAN/S/B/I   $LPT/L ERROR/E NEWTE.FR
FORTRAN/S/B/I   $LPT/L ERROR/E NOACK.FR
FORTRAN/S/B/I   $LPT/L ERROR/E NOGYRO.FR
```

```
MAC              $LPT/L ERROR/E OEBL.SR
MAC              $LPT/L ERROR/E OLNM.SR
FORTRAN/S/B/I    $LPT/L ERROR/E OLT.FR
FORTRAN/S/B/I    $LPT/L ERROR/E OLTCK.FR
FORTRAN/S/B/I    $LPT/L ERROR/E OPRDPHZ.FR
FORTRAN/S/B/I    $LPT/L ERROR/E OVERRIDE.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P01A.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P01B.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P01C.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P01D.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P02A.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P02B.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P02C.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P03.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P04A.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P04B.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P04C.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P04D.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P05.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P05SCH.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P06.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P07A.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P07B.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P07C.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P08.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P09A.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P09B.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P10A.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P10B.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P10C.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P10D.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P11A.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P12A.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P12B.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P12C.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P13A.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P13B.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P13C.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P14A.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P14B.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P14SCH.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P15A.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P15BC.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P15SCH.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P16.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P17A.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P17B.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P17SCH.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P18.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P19A.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P19B.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P1AC.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P1AZLR.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P1DIS.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P1END.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P1INIT.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P1PRM.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P1RAD.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P1SEG.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P1TXT.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P1VDC.FR
```

```
FORTRAN/S/B/I    $LPT/L ERROR/E P1WAI.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P2FRZ.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P23SUB.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P2RNSTOP.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P2RUN.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P3BSUP.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P3PBLK.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P3RUN.FR
FORTRAN/S/B/I    $LPT/L ERROR/E P3TRM.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PANEL.FR
MAC              $LPT/L ERROR/E PANPARAM.SR PANOUT.SR PANOUT.RB/B
MAC              $LPT/L ERROR/E LOCOSYM.SR/S PATCH.SR
FORTRAN/S/B/I    $LPT/L ERROR/E PATCK.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PB23SUB.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PCHK.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PERRCHK.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PEXCAM.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PHAZ23.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PHOSCH.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PHZ1.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PI00.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PI01.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PI02.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PI03.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PI04.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PI05.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PI06.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PI07.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PI08.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PI09.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PI10.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PI11.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PI12.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PI13.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PI14.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PI15.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PI16.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PI17.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PI18.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PI19.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PICKY.FR
MAC              $LPT/L ERROR/E PANPARAM.SR PINDR.SR PINDR.RB/B
MAC              $LPT/L ERROR/E PKNM.SR
MAC              $LPT/L ERROR/E PLACE.SR
FORTRAN/S/B/I    $LPT/L ERROR/E PLTASSUMES.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PLTCOPIEDN.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PLTDECIDES.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PLTWAVESHI.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PMCAM.FR
MAC              $LPT/L ERROR/E PMCLR.SR
FORTRAN/S/B/I    $LPT/L ERROR/E PMINT.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PMOLT.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PMS.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PMSCHD.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PMWAV.FR
FORTRAN/S/B/I    $LPT/L ERROR/E POSADH.FR
FORTRAN/S/B/I    $LPT/L ERROR/E POSOLT.FR
FORTRAN/S/B/I    $LPT/L ERROR/E POSROG.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PPANEL.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PRHELP.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PRNTIT.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PRSUS.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PSPCH.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PSPEC.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PST1.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PSUS.FR
```

```
FORTRAN/S/B/I    $LPT/L ERROR/E PTURN.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PULLRANGE.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PUTSCORES.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PUTWIND.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PWAVE.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PZ23.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PZ3B.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PZDEMO.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PZEC.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PZERR.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PZREG.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PZSCREEN.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PZSEL.FR
FORTRAN/S/B/I    $LPT/L ERROR/E PZTXT.FR
FORTRAN/S/B/I    $LPT/L ERROR/E RADAR.FR
FORTRAN/S/B/I    $LPT/L ERROR/E RADOUT.FR
FORTRAN/S/B/I    $LPT/L ERROR/E RDACT.FR
FORTRAN/S/B/I    $LPT/L ERROR/E RDBUFF.FR
FORTRAN/S/B/I    $LPT/L ERROR/E RDERR.FR
MAC              $LPT/L ERROR/E RDFRAZ.SR FRAZSYM.SR
FORTRAN/S/B/I    $LPT/L ERROR/E RDRPLY.FR
FORTRAN/S/B/I    $LPT/L ERROR/E RDTILNOTCO.FR
FORTRAN/S/B/I    $LPT/L ERROR/E REMSEL.FR
FORTRAN/S/B/I    $LPT/L ERROR/E REPLAY.FR
FORTRAN/S/B/I    $LPT/L ERROR/E RESPOND.FR
FORTRAN/S/B/I    $LPT/L ERROR/E REXPLAIN.FR
FORTRAN/S/B/I    $LPT/L ERROR/E RLDIR.FR
FORTRAN/S/B/I    $LPT/L ERROR/E RNOCAL.FR
FORTRAN/S/B/I    $LPT/L ERROR/E RNOSCHD.FR
FORTRAN/S/B/I    $LPT/L ERROR/E ROGER.FR
MAC              $LPT/L ERROR/E RPCLOK.SR
FORTRAN/S/B/I    $LPT/L ERROR/E RPFOR.FR
FORTRAN/S/B/I    $LPT/L ERROR/E RPHEAD.FR
FORTRAN/S/B/I    $LPT/L ERROR/E RPINITAC.FR
FORTRAN/S/B/I    $LPT/L ERROR/E RPKEY.FR
FORTRAN/S/B/I    $LPT/L ERROR/E RR1FIN.FR
FORTRAN/S/B/I    $LPT/L ERROR/E RSB.FR
FORTRAN/S/B/I    $LPT/L ERROR/E RTINIT.FR
FORTRAN/S/B/I    $LPT/L ERROR/E RTZEC.FR
FORTRAN/S/B/I    $LPT/L ERROR/E RUNIT.FR
FORTRAN/S/B/I    $LPT/L ERROR/E RUNKILL.FR
FORTRAN/S/B/I    $LPT/L ERROR/E RUNSTOP.FR
FORTRAN/S/B/I    $LPT/L ERROR/E RZEC.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SAYIT.FR
MAC              $LPT/L ERROR/E SBF.SR
FORTRAN/S/B/I    $LPT/L ERROR/E SC1214.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SC1518.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SC19.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SC35.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SC68.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SC911.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SCHINIT.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SCHREAD.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SCHWRITE.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SCORE.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SDIGIT.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SELBUT.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SELECT.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SELNV.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SFSET.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SGNOFF.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SGOODKY.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SHEAD.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SHFSTOP.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SHUFFLE.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SINON.FR
```

```
FORTRAN/S/B/I    $LPT/L ERROR/E SLURP.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SMISH.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SMOTHR.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SMREC.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SPBUF.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SPDMP.FR
MAC              $LPT/L ERROR/E SPDR.SR
FORTRAN/S/B/I    $LPT/L ERROR/E SPEAKPILOT.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SPIN.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SPOUT.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SR1FIN.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SR1ST.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SRMON.FR
FORTRAN/S/B/I    $LPT/L ERROR/E START1.FR
FORTRAN/S/B/I    $LPT/L ERROR/E STHELP.FR
FORTRAN/S/B/I    $LPT/L ERROR/E STOPTURN.FR
FORTRAN/S/B/I    $LPT/L ERROR/E STPILOT.FR
FORTRAN/S/B/I    $LPT/L ERROR/E STSK.FR
FORTRAN/S/B/I    $LPT/L ERROR/E STUDTALK.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SUBMODIFY.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SUCOVFLO.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SUCPH.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SUGYRO.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SUMPUT.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SURPLY.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SUS.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SUSEND.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SUSHAN.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SUSTRM.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SUSWRITE.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SVT.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SWIND.FR
FORTRAN/S/B/I    $LPT/L ERROR/E SYSINIT.FR
FORTRAN/S/B/I    $LPT/L ERROR/E TABKOUT.FR
FORTRAN/S/B/I    $LPT/L ERROR/E TFB.FR
FORTRAN/S/B/I    $LPT/L ERROR/E TCT50.FR
FORTRAN/S/B/I    $LPT/L ERROR/E THINKPILOT.FR
FORTRAN/S/B/I    $LPT/L ERROR/E TIMCAL.FR
FORTRAN/S/B/I    $LPT/L ERROR/E TIMEOUT.FR
FORTRAN/S/B/I    $LPT/L ERROR/E TIMER.FR
FORTRAN/S/B/I    $LPT/L ERROR/E TIMSCHD.FR
FORTRAN/S/B/I    $LPT/L ERROR/E TOWER.FR
FORTRAN/S/B/I    $LPT/L ERROR/E TRN.FR
FORTRAN/S/B/I    $LPT/L ERROR/E TSKERRDLY1.FR
FORTRAN/S/B/I    $LPT/L ERROR/E TURN.FR
FORTRAN/S/B/I    $LPT/L ERROR/E TZEC.FR
FORTRAN/S/B/I    $LPT/L ERROR/E VARIMOD.FR
FORTRAN/S/B/I    $LPT/L ERROR/E VOICTST.FR
FORTRAN/S/B/I    $LPT/L ERROR/E VSOUT.FR
FORTRAN/S/B/I    $LPT/L ERROR/E VSPRES.FR
FORTRAN/S/B/I    $LPT/L ERROR/E WALOFF.FR
FORTRAN/S/B/I    $LPT/L ERROR/E WAVE.FR
FORTRAN/S/B/I    $LPT/L ERROR/E WHEELS.FR
FORTRAN/S/B/I    $LPT/L ERROR/E WIND.FR
FORTRAN/S/B/I    $LPT/L ERROR/E WOCK.FR
MAC              $LPT/L ERROR/E WRFRAZ.SR FRAZSYM.SR
FORTRAN/S/B/I    $LPT/L ERROR/E WRMES.FR
FORTRAN/S/B/I    $LPT/L ERROR/E YORN.FR
FORTRAN/S/B/I    $LPT/L ERROR/E ZTIM.FR
```

```
/CP2.MC
/MACRO TO COMPILE CPU2 ROUTINES
DELETE ERROR
FORTRAN/S/B/I $LPT/L ERROR/E BEGIN.FR
FORTRAN/S/B   $LPT/L ERROR/E BLOCK2.FR
FORTRAN/S/B   $LPT/L ERROR/E BLOCKF.FR
FORTRAN/S/B/I $LPT/L ERROR/E CHANGE.FR
FORTRAN/S/B/I $LPT/L ERROR/E CKCMN.FR
MAC           $LPT/L ERROR/E CLOK2.SR
MAC           $LPT/L ERROR/E CLOKF.SR
FORTRAN/S/B/I $LPT/L ERROR/E COLLECT.FR
FORTRAN/S/B/I $LPT/L ERROR/E CREATE.FR
MAC           $LPT/L ERROR/E PARTITION.SR DPART2.SR
FORTRAN/S/B/I $LPT/L ERROR/E FADOFF.FR
FORTRAN/S/B/I $LPT/L ERROR/E FORMIT.FR
FORTRAN/S/B/I $LPT/L ERROR/E FREETOWRCMN.FR
FORTRAN/S/B/I $LPT/L ERROR/E FRZOT.FR
MAC           $LPT/L ERROR/E GOBBLE.SR
FORTRAN/S/B/I $LPT/L ERROR/E GOOF.FR
FORTRAN/S/B/I $LPT/L ERROR/E GRESP2.FR
FORTRAN/S/B/I $LPT/L ERROR/E HEARSAY.FR
FORTRAN/S/B/I $LPT/L ERROR/E HELLO.FR
MAC           $LPT/L ERROR/E HSCDR.SR
FORTRAN/S/B/I $LPT/L ERROR/E IMAGES.FR
FORTRAN/S/B/I $LPT/L ERROR/E INIT2RT.FR
FORTRAN/S/B/I $LPT/L ERROR/E IPBIN2.FR
MAC           $LPT/L ERROR/E IPBOUT2.SR
FORTRAN/S/B/I $LPT/L ERROR/E LEVEL.FR
MAC           $LPT/L ERROR/E LOCU.SR CRAZY.SR
FORTRAN/S/B/I $LPT/L ERROR/E LOKFORWARD.FR
MAC           $LPT/L ERROR/E LOOKATME.SR
FORTRAN/S/B/I $LPT/L ERROR/E LOOKOUT.FR
MAC           $LPT/L ERROR/E OEBL.SR
FORTRAN/S/B/I $LPT/L ERROR/E OKTOUSEMEGATEK.FR
MAC           $LPT/L ERROR/E OLNM.SR
FORTRAN/S/B/I $LPT/L ERROR/E PICUP.FR
MAC           $LPT/L ERROR/E PKNM.SR
FORTRAN/S/B/I $LPT/L ERROR/E PLATEXT.FR
FORTRAN/S/B/I $LPT/L ERROR/E PRESENT.FR
FORTRAN/S/B/I $LPT/L ERROR/E S7BUG.FR
FORTRAN/S/B/I $LPT/L ERROR/E SAID.FR
FORTRAN/S/B/I $LPT/L ERROR/E SERVO.FR
FORTRAN/S/B/I $LPT/L ERROR/E SERVUP.FR
FORTRAN/S/B/I $LPT/L ERROR/E SETIT.FR
FORTRAN/S/B/I $LPT/L ERROR/E SFORMIT.FR
FORTRAN/S/B/I $LPT/L ERROR/E SKBRD.FR
FORTRAN/S/B/I $LPT/L ERROR/E SKPRO.FR
MAC           $LPT/L ERROR/E SLOWJOY.SR
FORTRAN/S/B/I $LPT/L ERROR/E SPEECH.FR
FORTRAN/S/B/I $LPT/L ERROR/E SPINIT.FR
FORTRAN/S/B/I $LPT/L ERROR/E START2.FR
FORTRAN/S/B/I $LPT/L ERROR/E STARTF.FR
FORTRAN/S/B/I $LPT/L ERROR/E STIFLE.FR
FORTRAN/S/B/I $LPT/L ERROR/E STOVERRIDE.FR
FORTRAN/S/B/I $LPT/L ERROR/E STUDSTATS.FR
FORTRAN/S/B/I $LPT/L ERROR/E SUSOFF.FR
FORTRAN/S/B/I $LPT/L ERROR/E SUSON.FR
FORTRAN/S/B/I $LPT/L ERROR/E S7BUG.FR
FORTRAN/S/B/I $LPT/L ERROR/E TALKOUT.FR
```

```
FORTRAN/S/B/I $LPT/L ERROR/E TERMINATE.FR
FORTRAN/S/B/I $LPT/L ERROR/E TEST.FR
FORTRAN/S/B/I $LPT/L ERROR/E TSKERRDLY.FR
FORTRAN/S/B/I $LPT/L ERROR/E TUNIT.FR
MAC           $LPT/L ERROR/E VSIFPHDR/S VALYZ.SR VALYZ.RB/B
MAC           $LPT/L ERROR/E VSIFPHDR/S VCHOS.SR VCHOS.RB/B
MAC           $LPT/L ERROR/E VSIFPHDR/S VCOMP.SR VCOMP.RB/B
MAC           $LPT/L ERROR/E VCORR.SR
FORTRAN/S/B/I $LPT/L ERROR/E VDC1VAL.FR
FORTRAN/S/B/I $LPT/L ERROR/E VDC2VAL.FR
FORTRAN/S/B/I $LPT/L ERROR/E VDCOFF.FR
FORTRAN/S/B/I $LPT/L ERROR/E VDCON.FR
FORTRAN/S/B/I $LPT/L ERROR/E VOIFP.FR
FORTRAN/S/B/I $LPT/L ERROR/E VOVRP.FR
MAC           $LPT/L ERROR/E VIFP.SR
MAC           $LPT/L ERROR/E VIN.SR
MAC           $LPT/L ERROR/E VIPDR.SR
MAC           $LPT/L ERROR/E VSIFPHDR/S VMAP.SR VMAP.RB/B
MAC           $LPT/L ERROR/E VSIFPHDR/S VOVEX.SR VOVEX.RB/B
FORTRAN/S/B/I $LPT/L ERROR/E VRPLD.FR
FORTRAN/S/B/I $LPT/L ERROR/E VRPRT.FR
FORTRAN/S/B/I $LPT/L ERROR/E VSPCL.FR
MAC           $LPT/L ERROR/E VICOMHDR/S MAIL2HDR/S VSIFPHDR/S VSRRC.SR VSRRC.RB/B
MAC           $LPT/L ERROR/E VUCLK.SR
MAC           $LPT/L ERROR/E VVUCL.SR
FORTRAN/S/B/I $LPT/L ERROR/E WNDCHG.FR
```

```
/DORICP.MC
/        - COMPILE MACRO FOR DORT (BOTH SIDES) ---
DELETE ERROR
FORTRAN/S ERROR/E BLOCKD.FR
FORTRAN/S ERROR/E CHOZ.FR
FORTRAN/S ERROR/E CONTACT1.FR
FORTRAN/S ERROR/E CONTACT2.FR
FORTRAN/S ERROR/E D$CREATE.FR
FORTRAN/S ERROR/E D$DONE.FR
MAC       ERROR/E PARTITION/S D$DPART.SR D$DPART.RB/B
FORTRAN/S ERROR/E D$GLBF.FR
MAC       ERROR/E D$GLIB.SR
MAC       ERROR/E D$PANPARAM.SR D$PANOUT.SR D$PANOUT.RB/B
MAC       ERROR/E D$PANPARAM.SR D$PINDR.SR D$PINDR.RB/B
MAC       ERROR/E D$RDFRAZ.SR FRAZSYM
MAC       ERROR/E D$SBF.SR
FORTRAN/S ERROR/E D$SPBUF.FR
FORTRAN/S ERROR/E D$SPDMP.FR
MAC       ERROR/E D$SPDR.SR
FORTRAN/S ERROR/E D$SPIN.FR
FORTRAN/S ERROR/E D$SPOUT.FR
FORTRAN/S ERROR/E D$TSKERRDLY1.FR
FORTRAN/S ERROR/E D$VSOUT.FR
FORTRAN/S ERROR/E D$GRESP.FR
FORTRAN/S ERROR/E D2$BLOCKDA.FR
FORTRAN/S ERROR/E D2$GRESP.FR
FORTRAN/S ERROR/E D2$OPTIONS.FR
FORTRAN/S ERROR/E D2$PLATEXT.FR
FORTRAN/S ERROR/E D2$STATS.FR
MAC       ERROR/E D2$VQADR.SR
FORTRAN/S ERROR/E DG2ATEST.FR
FORTRAN/S ERROR/E DIGITEST.FR
FORTRAN/S ERROR/E DORT1.FR
FORTRAN/S ERROR/E DORT2.FR
FORTRAN/S ERROR/E DORTIPB1.FR
FORTRAN/S ERROR/E DPLATEXT.FR
FORTRAN/S ERROR/E HELP.FR
MAC       ERROR/E HSC6050.SR
FORTRAN/S ERROR/E HSCTEST.FR
MAC       ERROR/E LOCO CZ1.SR LOCO1.RB/B
MAC       ERROR/E LOCO CZ2.SR LOCO2.RB/B
FORTRAN/S ERROR/E MEBATEST.FR
FORTRAN/S ERROR/E NEXT.FR
FORTRAN/S ERROR/E OPTIONS.FR
FORTRAN/S ERROR/E PANDARK.FR
FORTRAN/S ERROR/E PANLIGHT.FR
FORTRAN/S ERROR/E PLAYBACK.FR
FORTRAN/S ERROR/E PNFLASH.FR
FORTRAN/S ERROR/E PN1TEST.FR
FORTRAN/S ERROR/E PN2TEST.FR
MAC       ERROR/E RANDU.SR
```

```
FORTRAN/S ERROR/E RECORD.FR
FORTRAN/S ERROR/E RECOUNT.FR
MAC       ERROR/E SOFT2HSC.SR
FORTRAN/S ERROR/E STATS.FR
FORTRAN/S ERROR/E STBACT.FR
FORTRAN/S ERROR/E STBEXP.FR
FORTRAN/S ERROR/E STBMAC.FR
FORTRAN/S ERROR/E STBRDOS.FR
FORTRAN/S ERROR/E STBS.FR
FORTRAN/S ERROR/E TOUCH.FR
FORTRAN/S ERROR/E UPDATE1.FR
FORTRAN/S ERROR/E UPDATE2.FR
FORTRAN/S ERROR/E VIPHSC.FR
FORTRAN/S ERROR/E VIPTEST.FR
FORTRAN/S ERROR/E VX1ATEST.FR
FORTRAN/S ERROR/E VX1TEST.FR
FORTRAN/S ERROR/E VX2ATEST.FR
FORTRAN/S ERROR/E VX2TEST.FR
FORTRAN/S ERROR/E WIRETAP.FR
MAC       ERROR/E D$WRFRAZ.SR FRAZSYM
FORTRAN/S ERROR/E XREAD.FR
```

APPENDIX E

LOAD MACROS

```
/G1LD.MC
/MACRO TO LOAD CPU1 ROUTINES
/
DELETE GCA.OL GCA.LS LOG.CM
LOG;GTOD;ENDLOG
/
RLDR/E 21/K 32/C GCA.SV/S GCA.LS/L ^
 START1 BLOCK1 PINDR PANOUT PANEL IPBIN1 ^
 RDERR RNGSCHD TIMSCHD EXEC PLACE DIE ERRTEST ^
 PMS PSUS PZEC ^
 LOOKATME 14000/N ^
 ^
 [GETNEXT GRESP P23SUB SCHREAD SCHWRITE YORN SR1ST GETANS RDTILNOTCO ACSET SR1FIN, ^
 SDIGIT RSB SUCOVFLG, ^
 SHEAD COMBO, ^
 BWIND PRSUS, ^
 SMISH SMREC, ^
 SMOTHR FILL SUCPH]/V ^
 ^
 [WIND PLTASSUMES NEWADVISOR PLTDECIDES PLTCOPIEDN, ^
 DEDUCETHEC SPEAKPILOT APRAX APREX, ^
 CONCEIVETHE PLTWAVESHI, ^
 MOVEPILOT]/V ^
 ^
 [SBF, ^
 RR1FIN TFB SUMPUT, ^
 DONE VSOUT WRFRAZ EX1PERT EXPERT PICKY, ^
 KPROC KTEACH KSTUD OEBL IADR DISPATCH IGOODKY SGOODKY SHFSTOP MODIFY, ^
 NEWTE SCHINIT, ^
 PRNTIT LIST FRREST FR301] ^
 ^
 [PB23SUB FTHSET, ^
 ACVERT SUSTRM SUSEND, ^
 SINON SGNOFF, ^
 PZREG, ^
 FEED, ^
 ENDFEED GO, ^
 TURN, ^
 TRN POSADH GTREND, ^
 EXPLAIN REXPLAIN RADOUT RDBUFF] ^
 ^
 [RTZEC, ^
 PHAZ23 SFSET PZTXT PZSCREEN, ^
 PZ23, ^
 MODELINIT VARIMOD PMSCHD PCHK, ^
 RUNIT APE1NIT, ^
 APE2NIT, ^
 APE3NIT APE4NIT APE5NIT, ^
 THINKPILOT RADAR LOOKUP RNGCAL, ^
 RPCLOK ATRPLY ERRHAN RPINITAC SURPLY SLURP, ^
 SCORE, ^
 CRSTUFE STHELP, ^
 FRDIALOG HED4, ^
 FR304, ^
 FR912 FR3HELP, ^
 OPRUN GAMOD OPRDPHZ PUTSCORES, ^
 INITRT] ^
```

```
^
[SYSINIT P1INIT P1END. ^
P3PBLK SELECT ADAPT P3BSUP. ^
PZSEL SELNV DESCRPROB. ^
REMSEL PZERR P1BEG P1TXT. ^
VOICTST LOGVT VSPRES DIGIN DWAIT RESPOND P1PRM P1VDC. ^
SUS ISAY ACTSUS ISABUF. ^
P1DIS P1AC F1ACINIT P1RAD P1WAI P1AZLR. ^
REPLAY ERLOOKUP ACTIVITY. ^
FOR1. ^
FOR2 HEAD2. ^
FOR3 HEAD3. ^
FOR4 WRMES GREAL FILNM. ^
RPFOR PRHELP RPHEAD RPKEY. ^
SC35 SC68. ^
SC911 SC1214 SC1518. ^
P19B SC19 FB19. ^
PI00 PI01 PI02 PI03 PI04 PI05 PI06 PI07 PI08 PI09. ^
PI10 PI11 PI12 PI13 PI14 PI15 PI16 PI17 PI18. ^
OVERRIDE. ^
SUBMODIFY] ^
^
[PZDEMO SRMON. ^
PHZ1 TIMER. ^
P2RUN P2RNSTOP P2FRZ. ^
P3RUN PZ3B RUNSTOP] ^
^
[STSK. ^
APENIT. ^
HOWFAR HOWHIGH NOGYRO. ^
MODWIND CLEAR. ^
CSOVER. ^
P3TRM] ^
^
[RDACT GETBUFF RLDIR. ^
P07A PWAVE PHOSCH PTURN PST1. ^
PPANEL PSPCH PSPEC. ^
GETDIR. ^
RDRPLY] ^
^
[DEMO PI19. ^
KREPLAY. ^
TOWER PUTWIND LOST IMOFF LOW. ^
CLRNC WAVE IGNORE GYROKILL ENDAPOP BEGDES CLRBUTX CONTOW CLREG WALOFF. ^
DECK FINCON STPILOT OLT WHEELS BEATIT GIMMIE HEYFEED. ^
NOACK PULLRANGE SUGYRO HOLD MILER MARKIT HEYTZEC STOPTURN MSGFILL. ^
CKIN CKCRP CKGPP. ^
TGT50 BUTX APGP. ^
STUDTALK. ^
PMINT. ^
SUBHAN. ^
P10A P17B P18 P10D. ^
P01A P01B P01C P01D P02A. ^
P07B PMCAM. ^
P07C. ^
WOCK. ^
P03 P14A DIRT. ^
AFDNA AFAPGP AFWC PATCK P10B P12A P12B P11A. ^
P10C P08 P17SCH P13B P13C. ^
P04A P04B P04C. ^
P06 P09A P09B. ^
P16 P02B P14B P15A P13A. ^
P05. ^
P15BC P02C P12C P15BCH PMWAV. ^
P04D PEXCAM HOCK. ^
```

```
CK120 CKACK CKAGP CKBD CKCHK CKCLR CKCN CKCOR CKCWO. ^
CKWO CKEZN CKFCP CKGMR CKZN3. ^
CKADH CKHDCOR CKHN CKHO CKICS CKK3 CKK5 CKLAA CKNGA CKOLT CKOVR. ^
CKP18 CKPAT CKPCLR CKRFR CKTB CKTLS. ^
CKRNG CKROM. ^
DHCK. ^
PMOLT PO5SCH P14SCH P17A OLTCK. ^
SHUFFLE MOVIT] ^
^
SELBUT ROGER POSOLT POSROG DESEL HOWNOW.
ERIN P19A PERRCHK PMCLR ^
LEVEL1 RUNKILL RTINIT ^
CLOK PATCH OLNM PKNM IPBOUT1 TASKOUT TSKERRDLY1 GOOF1 IKBRD MENU IVT SVT ^
SPOUT SPIN SPDR SPBUF SPDMP ^
GLIB GLBF RDFRAZ SAYIT HOSAY MSGPICKED ^
SUSWRITE ACTOUT ACDMP ^
TZEC TIMEOUT ZTIM RZEC APEX TIMCAL ^
LOCO1 LOCO2 LOCO3 DPART @TFLIB@
/
MESSAGE LOAD BEGAN AT--
TYPE LOG.CM
MESSAGE LOAD FINISHED AT--
GTOD
BLEEP
```

```
/G2LD.MC
/MACRO TO LOAD CPU2 BACKGROUND ROUTINES
DELETE CTSB.LS CTSB.OL
/
RLDR/E 15/K 16/C CTSB.SV/S CTSB.LS/L ^
 START2 BLOCK2 OLNM ^
 IPBIN2 IPBOUT2 LOOKATME ^
 6000/N VIN ^
 ^
 [VRPRT, ^
 VCOMP VCHOS VALYZ, ^
 VSPCL]/V ^
 ^
 [TEST, ^
 VDC2VAL, ^
 FORMIT] ^
 ^
 [TUNIT SPINIT, ^
 SUSON SUSOFF SAID HEARSAY, ^
 VDCON VDCOFF COLLECT VGIFP VIFP SFORMIT VGVRP PRESENT FRZOT LEVEL BEGIN TERMINATE] ^
 ^
 [HELLO PKNM, ^
 STIFLE, ^
 INIT2RT LOGRT OKRT, ^
 STUDSTATS, ^
 STOVERRIDE, ^
 SKPRO, ^
 PLATEXT] ^
 ^
 CLOK2 VUCLK VVUCL DEBL GOOF ^
 HSCDR VIPDR SPEECH VDC1VAL VRPLD VSRRC VOVEX VMAP VCORR ^
 TALKOUT LOOKOUT LOWFORWARD FREETOWRCMN TSKERRDLY CKCMN  ^
 SKBRD GRESP2 ^
 DPART2 LOCO @TFLIB@
/
```

```
/GFLD.MC
/MACRO TO LOAD CPU2 FOREGROUND ROUTINES
DELETE CTSF.LS
RLDR/D/L 2/K CTSF.SV/S CTSF.LS/L  ^
 STARTF BLOCKF GOBBLE CLOKF TSKERRDLY OKTOUSEMEGATER ^
 IMAGES SETIT CHANGE FADOFF PICUP CREATE WNDCHG SERVO SERVUP ^
 SLOWJOY S/BUG GRAPHICS.LB @IFLIB@
SYMBL CTSF.SV
```

```
/         LOAD MACRO FOR DORT1.SV (SIDE 1 OF DORT FOR GCA) ---
DELETE DORT.LS DORT1.OL

RLDR/D/E DORT.LS/L 20/K 32/C  DORT1 BLOCKD ^
6000/N  ^
D$SBF ^
PN1TEST D$PINDR D$PANOUT ^
D$GLIB D$GLBF D$VSOUT D$TSKERRDLY1 ^
VX1TEST VX1ATEST ^
OPTIONS CONTACT2 XREAD ^
PNFLASH RECOUNT ^
UPDATE1 [D$DONE,STBEXP,STBACT]/V D$RDFRAZ D$WRFRAZ DPLATEXT D$GRESP NEXT HELP ^
STAT3 DORTIPB1 LOC01 ^
D$SPIN D$SPOUT D$SPDMP D$SPBUF D$SPDR ^
WIRETAP RECORD PLAYBACK ^
PANLIGHT PANDARK ^
D$DPART RTFLIB@


SYMBL DORT1.SV
```

APPENDIX F

CROSS-REFERENCE OF
TIME AND RANGE SCHEDULED ROUTINES AND THEIR CALLERS,
MAILBOXES AND EVENTS

Routine Time Scheduled	Caller
BEGDES	APGP
BUTX	TGT50
CK120	P05
CKACK	PSPEC
CKBD	CKAGP
CKCHK	PSPEC
CKCN	PSPEC
CKCOR	P06
CKCWO	PPANEL
CKFCP	P11A
CKGMR	P01B
CKHDCOR	P14SCH
CKHN	P02C
CKHO	PHOSCH
CKICS	PPANEL
CKK3	P02A, P02B
CKK5	PPANEL
CKLAA	PSPEC
CKMGR	P01B
CKOLT	PSPEC
CKOVR	OLTCK
CKPAT	P12A
CKRFR	P12C
CKWO	P15SCH
CKZN3	P05SCH
CLRBUTX	CONTOW
CLRNC	TOWER
CONTOW	OLT
FINCON	GIMMIE, BUTX
GIMMIE	BUTX
HEYFEED	BUTX
HEYTZEC	CLRBUTX, ENDFEED, FEED, MODELINIT
HOLD	FEED, BUTX, ENDFEED, HOSAY, SAYIT
IGNORE	TOWER
IMOFF	PLTWAVESHI, LOST, OLT, CONCEIVETHE
LOST	RADAR
LOW	MOVEPILOT
MSGFILL	MSGPICKED

Routine Time Scheduled	Caller
NOACK	APGP
P05	CK120
P17A	P17SCH
STOPTURN	PICKY
STPILOT	FEED
TOWER	PANEL, CLREQ
WALOFF	WAVE
WAVE	TOWER
WHEELS	FINCON

Routine Range Scheduled	Caller
BEATIT	MODELINIT
CKADH	PI09
CKCLR	P10A
CKCWO	P10A
CKNGA	P13A
CKP18	PI18
CKPCLR	PI10
CKRNG	P08, CKRNG
CKROM	CKRNG
CKTB	PI17
CLREQ	MODELINIT
DECK	MODELINIT
ENDAPGP	APGP
GYROKILL	P23SUB
MARKIT	PMSCHD
MILER	PMSCHD
OLT	MODELINIT
PHOSCH	PI12
PULLRANGE	MODELINIT

Mailboxes

Mailbox	Received By	Transmitted From
BXACT	ACTIVITY	RPCLOK
	ATTRPLY	RPCLOK
BXCOG	SAID	VSPCL
	TEST	VVUCL, STIFLE
	VDC2VAL	VVUCL, STIFLE
BXCYC	APEX	CLOK, RUNKILL
	APRAX	CLOK, RUNKILL
	APREX	CLOK, RUNKILL
BXFED	ENDFEED	HOLD, SAYIT
	FEED	GIMMIE, HEYFEED, HOLD, SAYIT
	HOSAY	HOLD, SAYIT
BXFZ1	DWAIT	RESPOND, TIMER
	P1WAI	P1AZLR, TIMER
BXPLY	SPBUF	SPDMP, SPIS, SPOUT
BXRC	SPDMP	SPIS
BXREC	VGIFP	VIPDR, VUCLK
	VSRRC	SUSOFF, TERMINATE, VIPDR
BXRPL	RADOUT	RPCLOK
	RDRPLY	RPCLOK
BXRZ	RZEC	APEX, APRAX, APREX
BXSPH	REPLAY	SPBUF
	RUNSTOP	SPDMP
BXTIM	STARTF	CLOKF

Events

Event	Wakeup Waited For By	Wakeup Generated By
EVERR	REPLAY	RPCLOK
EVEXPL	EXPLAIN	REXPLAIN
EVKEY	GETANS	KSTUD
	P1AZLR	KSTUD
	RESPOND	KSTUD
	REXPLAIN	KSTUD
	TIMEOUT	KSTUD
	IKBRD	KTEACH
	PZREQ	KTEACH
	START1	KTEACH
	TEST	SKBRD
	VDC2VAL	SKBRD
EVKYST	INITRT	INIT2RT (through IPBIN1)
EVPHZ	P1AC	HEYTZEC
	P2RUN	HEYTZEC, KTEACH, HOLD, PERRCHK
	P3RUN	HEYTZEC, KTEACH, HOLD
	PZ3B	HEYTZEC, KTEACH, HOLD
	PZDEMO	HEYTZEC, KTEACH, HOLD
	SR1FIN	RR1FIN
	P23SUB	STARTF (sent through CKCMN and IPBIN1)
	ISAY	SPEECH (through IPBIN1)
	P1VDC	SPEECH (through IPBIN1)
	PB23SUB	SPEECH (through IPBIN1)
	SUSTRM	SPEECH (through IPBIN1)
	VOICTST	SPEECH (through IPBIN1)

Events (Cont)

Event	Wakeup Waited For By	Wakeup Generated By
EVPHZ (Cont)	INIT2RT	SPEECH
	OVERRIDE	STOVERRIDE
	KREPLAY	REPLAY
	P3TRM	REPLAY
EVSPN	RESPOND	SPDMP
EVSPT	DIGIN	SPBUF
	DONE	SPBUF
	P1AZLR	SPBUF
	P1PRM	SPBUF
EVSTP	VOICTST	SVT
	P1VDC	SVT
EVTXT	P1PRM	PLATEXT (through IPBIN1)
	PZTXT	PLATEXT (through IPBIN1)
	REXPLAIN	STUDSTATS (through IPBIN1)
EVVIN	ISAY	PINDR
	LEVEL1	PINDR
	P1AZLR	PINDR
	SAYIT	PINDR
EVVRO	DONE	PINDR
	P1AZLR	PINDR
EVVRPD	SUSON	VRPLD
	VDCON	VRPLD
EVVST	ISAY	PINDR
EVZEC	STSK	INITRT, MODIFY, NEWTE, PRNTIT, OKRT, OVERRIDE, ZTIM

APPENDIX G

INTERPROCESSOR BUS IDENTIFICATIONS

CPU 1 ID's

INDEX	SOURCE	DESTINATION	NO. OF ARGUMENTS	PURPOSE
1-IDMEGSTR	PHASE1,REPLAY	IMAGES	UP TO 41	MEGATEK STRING
2-IDIMAGES	PHAZ*	IMAGES	2	DISPLAY CONTROL
3-IDPICUP	RADAR,PHAZ*	PICUP	7	A/C UPDATE
4-IDPKSRV	REPLAY	PICUP,SERVO	10	RADAR REPLAY
5-IDSERVO	RADAR	SERVO	3	SERVO UPDATE
6-IDTEXT	PHAZ*,REPLAY	PLAYTEXT	7	TEXT FILE
7-IDSTIFLE	KSTUD,KTEACH	STIFLE	1	STOP VOICE TEXT
8-IDTIME	TZEC	IPBIN2	1	CLOCK SYNC
9-IDSPEECH	PHAZ*	SPEECH	9	START SPEECH
10-IDLEVEL	PHAZ*	LEVEL	1	START LEVEL
11-IDHEARSAY	ISAY	HEARSAY	8	START HEARSAY
13-IDDIE	PHAZ*	TALKOUT	1	KILL CPU 2, SYNC FGND
14-IDSKPRO	KSTUD	SKPRO	1	START SKPRO
15-IDPRESENT	INITRT	PRESENT	7	START PRESENT
16-IDMENU	MENU	SKPRO	2	DISPLAY MENU
17-IDSTUDSTATS	KSTUD	STUDSTATS	1	START STUDSTATS
18-IDKILLTSK	ALL	TALKOUT	1	DESTROY BY ID
19-IDSKBRD	KEYBOARD	SKBRD	1	START SKBRD
20-IDHELLO	KSTUD	HELLO	1	START HELLO
21-IDINITRT	INITRT	INITRT	1	START NEW R/T
22-IDCRT	PHAZ*	IPBIN2	UP TO 41	CRT STRING
23-IDOVERRIDE	OVERRIDE	STOVER	8	START STOVERRIDE
24-IDFF	MANY PLACES	IPBIN2	1	ERASE STUDENT CRT

CPU 2 ID's

INDEX	SOURCE	DESTINATION	NO. OF ARGUMENTS	PURPOSE
1-IDKPROC	SKBRD	KPROC	2	KEYBOARD PROCESSING
2-IDSUS	SAID	SUS	7	SPEECH UNDERSTANDING
3-IDNSPRES	PRESENT	VSPRES	8	SPEECH PROMPTING
5-IDRADAR	SERVO	RADAR	2	SERVO UPDATES
7-IDAWAKE	ALL	ALL	1	WAKEUP ON EVENT
8-IDSINON	HELLO	SINON	4	TRAINEE IDENTIFICATION
9-IDLEVEL1	VDC	LEVEL1	1	T/E SPEECH INPUT LEVEL
10-IDRDCHG	SETIT	IPBIN1	1	RADAR DATA UPDATE
11-IDKILL	ALL	TASKOUT	1	KILL TASKS
12-IDOVERRIDE	STOVER	TASKOUT	9	OVERRIDE TASK INFO

APPENDIX H

CROSS-REFERENCE OF COMMON VARIABLES

This appendix consists of separately bound computer listings which provide an alphabetic listing of common variables, with the names of all routines that reference and/or modify each variable.

APPENDIX I

GLOSSARY OF AIRCRAFT/PILOT/ENVIRONMENTAL (APE) LOCAL VARIABLES

Variable Name	Type	Indexed By	Dimension
ADVHDG	Integer		Degrees magnetic

Heading appearing in, or accompanying, GCA advisory currently being processed by APE.

Variable Name	Type	Indexed By	Dimension
ADVID	Integer		Dimensionless

Phrase number of GCA advisory currently being processed by APE.

Variable Name	Type	Indexed By	Dimension
ALPHA	Real		Dimensionless

At time t the simulated pilot conceives that if he maintains a rate-of-descent of E_{old} feet-per-second his aircraft will fly parallel to the glidepath. Associated with this estimate of the correct rate-of-descent is variance V_{old}. At time $t+\Delta t$ the pilot copies a GCA advisory which causes him to conceive a new estimate E_{new} of the correct rate-of-descent, with associated variance V_{new}. Assuming that the true value of correct rate-of-descent, Y_I, is of the form:

$$Y_I = E_{old} + (1-\alpha)E_{new}$$

Then the variance associated with Y_I will be lowest if:

$$\alpha = \text{ALPHA} = \frac{V_{new}}{V_{new} + V_{old}}$$

Variable Name	Type	Indexed By	Dimension
BASMAX(5) BASMIN(5)	Real Array	PTYP	Knots

A pilot of skill level I, while attempting to maintain an airspeed of X knots, will actually cause his aircraft to display a range of airspeeds bounded by X+BASMIN(I) and X+BASMAX(I), and pseudo-normally distributed about a mean airspeed equal to the average of X+BASMIN(I) and X+BASMAX(I).

Variable Name	Type	Indexed By	Dimension
BHDMAX(5) BHDMIN(5)	Real Array	PTYP	Degrees/second

A pilot of skill level I, while attempting to maintain a rate-of-turn of X degrees/second, will actually cause his aircraft to display a range of rates-of-turn bounded by X+BHDMIN(I) and X+BHDMAX(I), and pseudo-normally distributed about a mean rate of turn equal to the average of X+BHDMIN(I) and X+BHDMAX(I).

BLIPHEIGHT BLIPSIZE	Real		Feet

This is the vertical distance between those two points in real space the radar images of which would coincide with the endpoints of the simulated aircraft radar image (target) currently appearing on the GCA-CTS simulated PAR display.

BYDMAX(5) BYDMIN(5)	Real Array	PTYP	Feet/minute

A pilot of skill level I, while attempting to maintain a rate of climb of X feet/minute, will actually cause his aircraft to display a range of rates-of-climb bounded by X+BYDMIN(I) and X+BYDMAX(I) and pseudo-normally distributed about a mean rate-of-climb equal to the average of X+BYDMIN(I) and X+BYDMAX(I).

DYVM(5)	Real Array	PTYP	Dimensionless

The simulated pilot; upon copying a new glidepath advisory (e.g., "above G/P") formulates, on the basis of the type of advisory and his current estimated distance from touchdown, an estimate of his current vertical distance from the glidepath. Associated with his estimate is an uncertainty/variance $V\Delta_y$. APE assumes that, all else being equal, the variance associated with a skillful pilot's altitude inference will be smaller than the variance associated with a less skillful pilot's altitude inference. Therefore APE models a pilot of skill level I as inferring his "true" altitude from a newly-copied advisory with an uncertainty DYVM(I) times as great as that of a pilot of skill level 1 under identical circumstances.

Variable Name	Type	Indexed By	Dimension
EDELY	Real		Feet

Pilot's current estimate of his vertical displacement above the glidepath (i.e., EDELY<0 implies the pilot estimates he is below the glidepath).

Variable Name	Type	Indexed By	Dimension
EVARDELY	Real		$Feet^2$

Variance associated with pilot's current estimate of his displacement, EDELY, above the glidepath.

Variable Name	Type	Indexed By	Dimension
EVARYDI	Real		$Feet^2/second^2$

Variance pilot associates with his estimate – based on the two most recently-copied glidepath advisories and corresponding altitude inferences, and his intervening indicated rate of descent, only! – of the correct rate of descent to maintain upon intercepting the glideslope.

Variable Name	Type	Indexed By	Dimension
EYDI	Real		Feet/second

Pilot's current estimate – based on the two most recently-copied glidepath advisories and corresponding altitude inferences, and his intervening indicated rate of descent, only! – of the correct rate of descent to maintain upon intercepting the glideslope.

Variable Name	Type	Indexed By	Dimension
FACTOR	Real		Dimensionless

In subroutines APREX and APRAX, equals the absolute value of the current vertical distance from the center of the displayed target in the (elevation, azimuth) display to the displayed (glideslope, course/centerline), expressed as a fraction of the current target width.

Variable Name	Type	Indexed By	Dimension
FMAX FMIN	Real		Feet

Variable Name	Type	Indexed By	Dimension
FNOCOPY(5)	Integer Array	PTYP	Dimensionless

A pilot of skill level I will fail to copy

$$100 \text{ X } 1 - \frac{\text{FNOCOPY (I)}}{32767}$$

percent of the glidepath advisories transmitted to him by the controller trainee and recognized/understood by SUS.

GAMMA	Real		Dimensionless

The quantity $(1-\alpha)$ as discussed under "ALPHA."

GP	Real		Feet

The elevation of the glidepath at the range corresponding to the current value in APREX of ACX.

NBUFCHECKD	Integer		Dimensionless

The number of SUS buffers which have been examined for the presence of new (as-yet-unprocessed-by-APE) GCA advisories so far in the current 0.5-second simulation cycle.

SA1AS(5) SA1YD(5) SA1HD(5)	Real Array	PTYP	Dimensionless

Tables of standard airspeed-, rate-of-climb-, and rate-of-turn-stability parameters. Values for stability parameters PTA1AS, PTA1YD, and PTA1HD appropriate to the pilot-type being simulated are extracted from these tables.

Variable Name	Type	Indexed By	Dimension
SCOAS(4)	Real Array	ACTYP	Knots
SCOYD(4)	Real Array	ACTYP	Feet/minute
SFAAS(4)	Real Array	ACTYP	Knots
SPTAS(4)	Real Array	ACTYP	Knots
SYDI(4)	Real Array	ACTYP	Feet/minute

Tables of Standard Climbout Airspeed, Climbout Rate-of-Climb, Final Approach Airspeed, Pattern Airspeed, and Initial Rate-of-Descent. Values for the corresponding "GCA flight-rules" parameters PTASCLO, PTYDCLO, PTASFA, PTASPAT appropriate to the aircraft type being simulated are extracted from these tables. Furthermore, SYDI (ACTYP) is the rate-of-descent which the simulated pilot first assumes to be the correct rate-of-descent upon copying the "Begin Descent" advisory.

SECPZ	Real		Seconds

In a restricted (elevation, azimuth) mode simulation the simulated aircraft radar image (target) will traverse (one-way) an elevation or azimuth zone on the average once each SECPZ seconds.

SVARYDI(5)	Real Array	PTYP	Feet^2/minute^2

A pilot of skill level I, upon copying the "Begin Descent" advisory, will initially attempt to achieve and maintain a rate-of-descent of SYDI(I) feet/minute. APE assumes, however, that the simulated pilot is aware that the value SYDI(I) (corresponding to the no-wind ideal rate-of-descent for a 3° glideslope for the given aircraft type's final approach airspeed) need not necessarily be the correct constant rate-of-descent for the current wind conditions. Therefore with the pilot's initial estimate SYDI(I) of the correct/ideal rate-of-descent is associated a variance SVARYDI (PTYP) – which in the current APE implementation is hard-coded as being pilot-type-independent, but need not be so coded – reflecting the pilot's uncertainty of the correct rate-of-descent.

APPENDIX J

ELEVATION AND AZIMUTH ZONE INTERPRETATION

Figure J1 shows the target division scheme used in GCA-CTS for elevation. Figure J2 shows the target division scheme for azimuth. The shaded areas are overlap zones in which either advisory is considered to be correct.

Table J1 shows the computations for glidepath target zone determinations, and Table J2 gives the pilot altitude error estimates associated with glidepath advisories. Table J3 gives the computations for course zone determinations. No course deviation error estimates are provided since the course position advisories do not cause course changes.

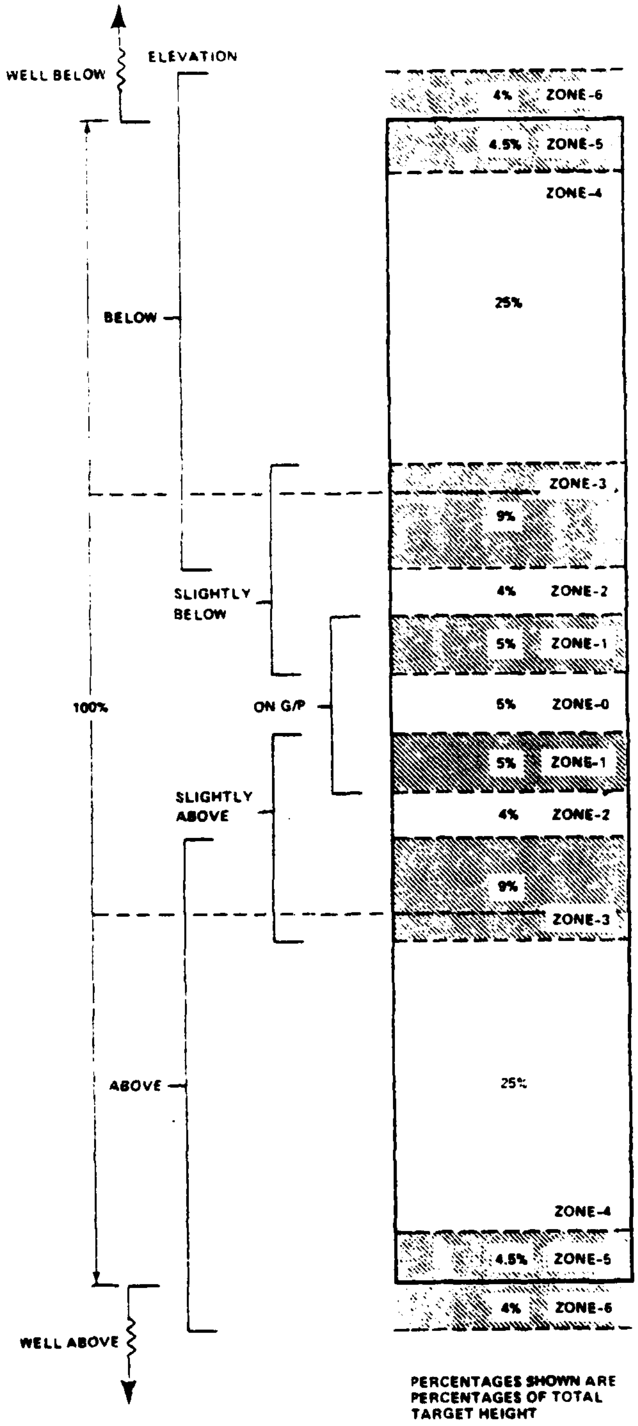


Figure J1. Description of Current Elevation Zone if Elevation Cursor Intersects Displayed Target in Indicated Region

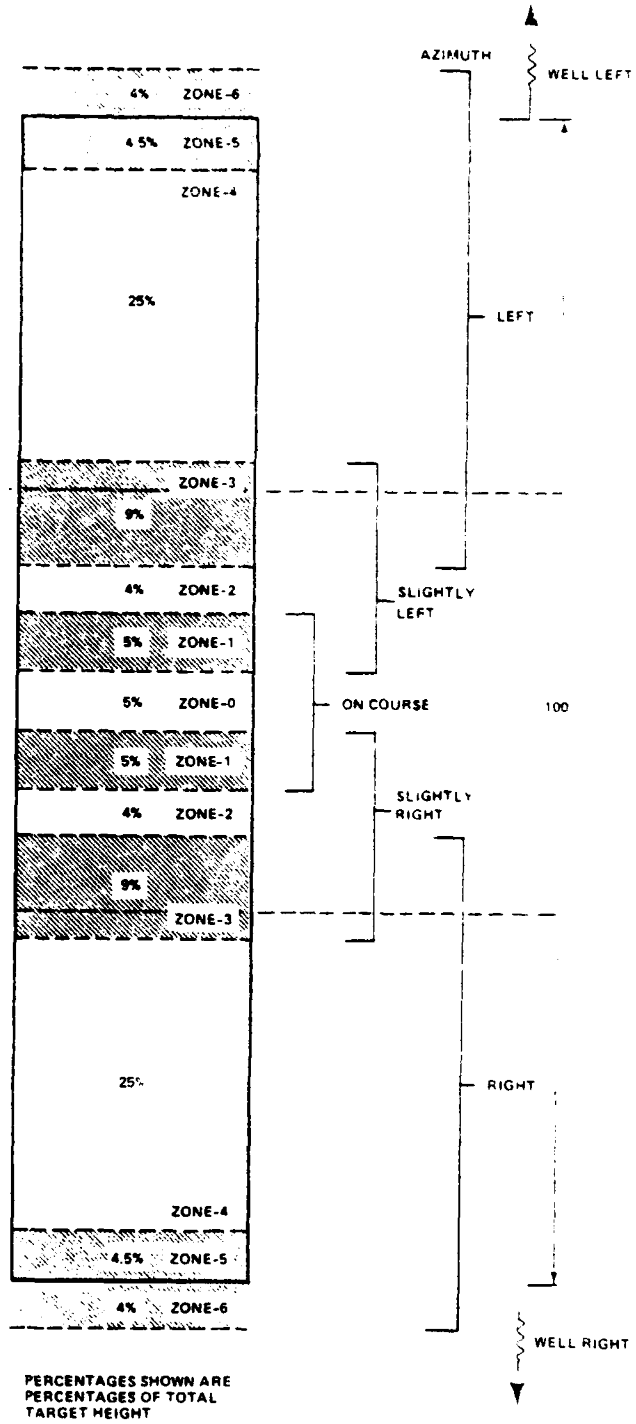


Figure J2. Description of Current Azimuth Zone if Azimuth Cursor Intersects Displayed Target in Indicated Region

TABLE J1. ELEVATION ZONE COMPUTATIONS

ZONE	AIRCRAFT POSITION DESCRIBED AS	AIRCRAFT AT ALTITUDE Y FEET RESIDES IN THIS ZONE IFF
-6	WELL BELOW G/P	$Y < GP - 0.52\beta$
-5	WELL BELOW OR BELOW G/P	$GP - 0.52\beta \leq Y < GP - 0.455\beta$
-4	BELOW G/P	$GP - 0.455\beta \leq Y < GP - 0.205\beta$
-3	BELOW OR SLIGHTLY BELOW G/P	$GP - 0.205\beta \leq Y < GP - 0.115\beta$
-2	SLIGHTLY BELOW G/P	$GP - 0.115\beta \leq Y < GP - 0.075\beta$
-1	ON OR SLIGHTLY BELOW G/P	$GP - 0.075\beta \leq Y < GP - 0.025\beta$
0	ON G/P	$GP - 0.025\beta < Y < GP + 0.025\beta$
1	ON OR SLIGHTLY ABOVE G/P	$GP + 0.025\beta < Y \leq GP + 0.075\beta$
2	SLIGHTLY ABOVE G/P	$GP + 0.075\beta < Y \leq GP + 0.115\beta$
3	ABOVE OR SLIGHTLY ABOVE G/P	$GP + 0.115\beta < Y \leq GP + 0.205\beta$
4	ABOVE G/P	$GP + 0.205\beta < Y \leq GP + 0.455\beta$
5	WELL ABOVE OR ABOVE G/P	$GP + 0.455\beta < Y \leq GP + 0.52\beta$
6	WELL ABOVE G/P	$GP + 0.52\beta < Y$

where $\beta = f(Z) = 2(Z+3605.07)(1.1345 \times 10^{-2})A$ (feet)

(target height in real space corresponding to a full elevation display blip)

and $GP = Z$ for $3° = .0524Z$ (feet)

and user selects A according to the rule:

$$A = \frac{\text{blip size desired for target at 9 n.m. from touchdown}}{1.5} \quad \text{(inches)}$$

β = BLIPSIZE, BLIPHEIGHT
Z = ACZ

TABLE J2. PILOT ALTITUDE ERROR ESTIMATE (Δy) ASSOCIATED WITH GLIDEPATH ADVISORIES

ADVISORY	PILOT ASSUMES
WELL BELOW	$\Delta y = -.455\beta$
BELOW	$\Delta y = -.3375\beta$
SLIGHTLY BELOW	$\Delta y = -.115\beta$
ON	$\Delta y = 0.0$
SLIGHTLY ABOVE	$\Delta y = +.115\beta$
ABOVE	$\Delta y = +.3375\beta$
WELL ABOVE	$\Delta y = +.455\beta$
COMING DOWN	No Assumption
COMING UP	No Assumption
GOING ABOVE	$\Delta y = +.05\beta$
GOING FURTHER ABOVE	$\Delta y = +.18\beta$
GOING BELOW	$\Delta y = -.05\beta$
GOING FURTHER BELOW	$\Delta y = -.18\beta$
BEGIN DESCENT	$\Delta y = 0$
APPROACHING G/P	No Assumption

Δy = pilot's impression of the current value of $(y - y_{g/p})$

(EDELY)

where $\beta = 2(Z+3605.07)(1.1345 \times 10^{-2})A$ (feet)

(BLIPSIZE, BLIPHEIGHT)

TABLE J3. AZIMUTH ZONE COMPUTATIONS

ZONE #	AIRCRAFT POSITION DESCRIBED AS	AIRCRAFT AT OFFSET OF X FEET RESIDES IN THIS ZONE IFF
-6	WELL LEFT OF COURSE	$X < -0.52\beta$
-5	WELL LEFT OR LEFT OF COURSE	$-0.52\beta \leq X < -0.455\beta$
-4	LEFT OF COURSE	$-0.455\beta \leq X < -0.205\beta$
-3	LEFT OR SLIGHTLY LEFT OF COURSE	$-0.205\beta \leq X < -0.115\beta$
-2	SLIGHTLY LEFT OF COURSE	$-0.115\beta \leq X < -0.075\beta$
-1	ON OR SLIGHTLY LEFT OF COURSE	$-0.075\beta \leq X < -0.025\beta$
0	ON COURSE	$-0.025\beta < X < -0.025\beta$
1	ON OR SLIGHTLY RIGHT OF COURSE	$0.025\beta < X \leq 0.075\beta$
2	SLIGHTLY RIGHT OF COURSE	$0.075\beta < X \leq 0.115\beta$
3	RIGHT OR SLIGHTLY RIGHT OF COURSE	$0.115\beta < X \leq 0.205\beta$
4	RIGHT OF COURSE	$0.205\beta < X \leq 0.455\beta$
5	WELL RIGHT OR RIGHT OF COURSE	$0.455\beta < X \leq 0.52\beta$
6	WELL RIGHT OF COURSE	$0.52\beta < X$

where $\beta = f(Z) = 2(Z+3605.07)(2.8231 \times 10^{-2})A$ (feet)

(target width in real space corresponding to a full azimuth display blip)

and user selects A according to the rule:

$$A = \frac{\text{blip size desired for target at 9 n.m. from touchdown}}{1.5} \quad \text{(inches)}$$

β = BLIPSIZE, BLIPHEIGHT
Z = ACZ

APPENDIX K

STACK PARTITIONS

CPU 1 STACK PARTITION TABLE

```
PARTITION          ;INITIALIZE THE TABLE

PARTITION 60 2     ,RDBUFF, TIMER
PARTITION 60 1     ,SPBUF
PARTITION 125 1    ;SPDMP
PARTITION 150 3    ;LEVEL1, ISAY, SUSWRITE, SAYIT, SRMON
PARTITION 250 2    ;IPBIN1, IKBRD
PARTITION 300 2    ;PANEL, TASKOUT

PARTITION          ;TERMINATE THE TABLE
```

CPU 2 BACKGROUND STACK PARTITION TABLE

```
PARTITION          ;INITIALIZE THE TABLE

PARTITION 100      ;HEARSAY, STIFLE
PARTITION 150 4    ;CKCMN, VRPLD, SAID, VSRRC
PARTITION 250      ;IPBIN2
PARTITION 300 3    ;TALKOUT, LOOKOUT, LOKFORWARD

PARTITION          ;TERMINATE THE TABLE
```

APPENDIX L

LOAD ON CALL CROSS-REFERENCE TABLES

CPU 1 CROSS-REFERENCE TABLE

N. LOCO	ACSET	OACSET	XACSET
N. LOCO	ACVERT	OACVERT	XACVERT
N. LOCO	ADAPT	OADAPT	XADAPT
N. LOCO	APENIT	OAPENIT	XAPENIT
N. LOCO	APE1NIT	OAPE1NI	XAPE1NI
N. LOCO	APE2NIT	OAPE2NI	XAPE2NI
N. LOCO	APE3NIT	OAPE3NI	XAPE3NI
N. LOCO	APE4NIT	OAPE4NI	XAPE4NI
N. LOCO	APE5NIT	OAPE5NI	XAPE5NI
N. LOCO	APGP	OAPGP	XAPGP
N. LOCO	APRAX	OAPRAX	XAPRAX
N. LOCO	APREX	OAPREX	XAPREX
N. LOCO	BEATIT	OBEATIT	XBEATIT
N. LOCO	BEGDES	OBEGDES	XBEGDES
N. LOCO	BUTX	OBUTX	XBUTX
N. LOCO	CKCRP	OCKCRP	XCKCRP
N. LOCO	CKGPP	OCKGPP	XCKGPP
N. LOCO	CKIN	OCKIN	XCKIN
N. LOCO	CLEAR	OCLEAR	XCLEAR
N. LOCO	CLRBUT	OCLRBUT	XCLRBUT
N. LOCO	CLREQ	OCLREQ	XCLREQ
N. LOCO	CLRNC	OCLRNC	XCLRNC
N. LOCO	COMBO	OCOMBO	XCOMBO
N. LOCO	CONCEIV	OCONCEI	XCONCEI
N. LOCO	CONTOW	OCONTOW	XCONTOW
N. LOCO	CRSTUFE	OCRSTUF	XCRSTUF
N. LOCO	CSOVER	OCSOVER	XCSOVER
N. LOCO	DECK	ODECK	XDECK
N. LOCO	DEDUCET	ODEDUCE	XDEDUCE
N. LOCO	DEMO	ODEMO	XDEMO
N. LOCO	DESCRP	ODESCRP	XDESCRP
N. LOCO	DIGIN	ODIGIN	XDIGIN
N. LOCO	DONE	ODONE	XDONE
N. LOCO	DWAIT	ODWAIT	XDWAIT
N. LOCO	ENDAPGP	OENDAPG	XENDAPG
N. LOCO	ENDFEED	OENDFEE	XENDFEE
N. LOCO	ERLOOKU	OERLOOK	XERLOOK
N. LOCO	EX1PERT	OEX1PER	XEX1PERT
N. LOCO	EXPLAIN	OEXPLAI	XEXPLAIN

```
N.LOCO   FEED     OFEED    XFEED
N.LOCO   FILL     OFILL    XFILL
N.LOCO   FINCON   OFINCON  XFINCON
N.LOCO   FOR1     OFOR1    XFOR1
N.LOCO   FOR2     OFOR2    XFOR2
N.LOCO   FOR3     OFOR3    XFOR3
N.LOCO   FOR4     OFOR4    XFOR4
N.LOCO   FRDIAL   OFRDIAL  XFRDIAL
N.LOCO   FRREST   OFRREST  XFRREST
N.LOCO   FR3HELP  OFR3HEL  XFR3HEL
N.LOCO   FR304    OFR304   XFR304
N.LOCO   FR912    OFR912   XFR912
N.LOCO   FTHSET   OFTHSET  XFTHSET
N.LOCO   F1ACINI  OF1ACIN  XF1ACINIT
N.LOCO   GAMOD    OGAMOD   XGAMOD
N.LOCO   GETBUFF  OGETBUF  XGETBUF
N.LOCO   GETANS   OGETANS  XGETANS
N.LOCO   GETDIR   OGETDIR  XGETDIR
N.LOCO   GETNEX   OGETNEX  XGETNEX
N.LOCO   GIMMIE   OGIMMIE  XGIMMIE
N.LOCO   GO       OGO      XGO
N.LOCO   GPRUN    OGPRUN   XGPRUN
N.LOCO   GRESP    OGRESP   XGRESP
N.LOCO   GTREND   OGTREND  XGTREND
N.LOCO   GYROK    OGYROK   XGYROK
N.LOCO   HED4     OHED4    XHED4
N.LOCO   HEYFEED  OHEYFE   XHEYFEED
N.LOCO   HEYTZEC  OHEYTZ   XHEYTZEC
N.LOCO   HOLD     OHOLD    XHOLD
N.LOCO   HOWFAR   OHOWFAR  XHOWFAR
N.LOCO   HOWHIGH  OHOWHIG  XHOWHIG
N.LOCO   IGNORE   OIGNORE  XIGNORE
N.LOCO   IMOFF    OIMOFF   XIMOFF
N.LOCO   INITRT   OINITRT  XINITRT
N.LOCO   ISAY     OISAY    XISAY
N.LOCO   KPROC    OKPROC   XKPROC
N.LOCO   KREPLA   OKREPLA  XKREPLA
N.LOCO   LOST     OLOST    XLOST
N.LOCO   LOW      OLOW     XLOW
N.LOCO   MARKIT   OMARKIT  XMARKIT
N.LOCO   MSGFILL  OMSGFIL  XMSGFILL
N.LOCO   MILER    OMILER   XMILER
N.LOCO   MODELIN  OMODEL   XMODEL
N.LOCO   MODIFY   OMODIFY  XMODIFY
N.LOCO   MODWIN   OMODWIN  XMODWIN
N.LOCO   MOVEPI   OMOVEPI  XMOVEPI
.END
```

```
;PART 2 OF LOAD ON CALL TABLE
;----------------------------

N.LOCO  NEWAD    ONEWAD   XNEWAD
N.LOCO  NEWTE    ONEWTE   XNEWTE
N.LOCO  NOACK    ONOACK   XNOACK
N.LOCO  NOGYRO   ONOGYRO  XNOGYRO
N.LOCO  OEBL     OOEBL    XOEBL
N.LOCO  OLT      OOLT     XOLT
N.LOCO  OPRDPHZ  OOPRDPH  XOPRDPH
N.LOCO  OVERRID  OOVERRI  XOVERRI
N.LOCO  PB23SU   OPB23SU  XPB23SU
N.LOCO  PHAZ23   OPHAZ23  XPHAZ23
N.LOCO  PHZ1     OPHZ1    XPHZ1
N.LOCO  PKNM     OPKNM    XPKNM
N.LOCO  PLTASS   OPLTASS  XPLTASS
N.LOCO  PLTCOP   OPLTCOP  XPLTCOP
N.LOCO  PLTDEC   OPLTDEC  XPLTDEC
N.LOCO  PLTWAV   OPLTWAV  XPLTWAV
N.LOCO  PMINT    OPMINT   XPMINT
N.LOCO  POSADH   OPOSADH  XPOSADH
N.LOCO  PRNTIT   OPRNTIT  XPRNTIT
N.LOCO  PRSUS    OPRSUS   XPRSUS
N.LOCO  PULLRAN  OPULLRA  XPULLRA
N.LOCO  PUTSCO   OPUTSCO  XPUTSCO
N.LOCO  PZDEMO   OPZDEMO  XPZDEMO
N.LOCO  PZERR    OPZERR   XPZERR
N.LOCO  PZREQ    OPZREQ   XPZREQ
N.LOCO  PZSCR    OPZSCR   XPZSCR
N.LOCO  PZSEL    OPZSEL   XPZSEL
N.LOCO  PZTXT    OPZTXT   XPZTXT
N.LOCO  PZ23     OPZ23    XPZ23
N.LOCO  PZ3B     OPZ3B    XPZ3B
N.LOCO  P1AC     OP1AC    XP1AC
N.LOCO  P1AZLR   OP1AZLR  XP1AZLR
N.LOCO  P1DIS    OP1DIS   XP1DIS
N.LOCO  P1END    OP1END   XP1END
N.LOCO  P1INIT   OP1INIT  XP1INIT
N.LOCO  P1PRM    OP1PRM   XP1PRM
N.LOCO  P1RAD    OP1RAD   XP1RAD
N.LOCO  P1SEQ    OP1SEQ   XP1SEQ
N.LOCO  P1TXT    OP1TXT   XP1TXT
N.LOCO  P1VDC    OP1VDC   XP1VDC
N.LOCO  P1WAI    OP1WAI   XP1WAI
N.LOCO  P2RUN    OP2RUN   XP2RUN
N.LOCO  P23SUB   OP23SUB  XP23SUB
N.LOCO  P3BSU    OP3BSU   XP3BSU
N.LOCO  P3PBLK   OP3PBLK  XP3PBLK
N.LOCO  P3RUN    OP3RUN   XP3RUN
N.LOCO  P3TRM    OP3TRM   XP3TRM
```

N. LOCO	RADAR	ORADAR	XRADAR
N. LOCO	RDACT	ORDACT	XRDACT
N. LOCO	RDRPLY	ORDRPL	XRDRPL
N. LOCO	RDTILNO	ORDTILN	XRDTILN
N. LOCO	REMSEL	OREMSEL	XREMSEL
N. LOCO	REPLAY	OREPLAY	XREPLAY
N. LOCO	RESPOND	ORESPON	XRESPOND
N. LOCO	RLDIR	ORLDIR	XRLDIR
N. LOCO	RNGCAL	ORNGCAL	XRNGCAL
N. LOCO	RPFOR	ORPFOR	XRPFOR
N. LOCO	RPINIT	ORPINIT	XRPINIT
N. LOCO	RR1FIN	ORR1FIN	XRR1FIN
N. LOCO	RSB	ORSB	XRSB
N. LOCO	RTZEC	ORTZEC	XRTZEC
N. LOCO	RUNIT	ORUNIT	XRUNIT
N. LOCO	SC35	OSC35	XSC35
N. LOCO	SC68	OSC68	XSC68
N. LOCO	SC911	OSC911	XSC911
N. LOCO	SC1214	OSC1214	XSC1214
N. LOCO	SC1518	OSC1518	XSC1518
N. LOCO	SC19	OSC19	XSC19
N. LOCO	SCHREAD	OSCHREA	XSCHREA
N. LOCO	SCHWRIT	OSCHWRI	XSCHWRI
N. LOCO	SCORE	OSCORE	XSCORE
N. LOCO	SDIGIT	OSDIGIT	XSDIGIT
N. LOCO	SELECT	OSELECT	XSELECT
N. LOCO	SGNOFF	OSGNOFF	XSGNOFF
N. LOCO	SHEAD	OSHEAD	XSHEAD
N. LOCO	SHUFFLE	OSHUFFL	XSHUFFLE
N. LOCO	SINON	OSINON	XSINON
N. LOCO	SMISH	OSMISH	XSMISH
N. LOCO	SMOTHR	OSMOTHR	XSMOTHR
N. LOCO	SMREC	OSMREC	XSMREC
N. LOCO	SPEAKP	OSPEAKP	XSPEAKP
N. LOCO	SRMON	OSRMON	XSRMON
N. LOCO	SR1FIN	OSR1FIN	XSR1FIN
N. LOCO	SR1ST	OSR1ST	XSR1ST
N. LOCO	STOPTU	OSTOPTU	XSTOPTU
N. LOCO	STPILOT	OSTPILO	XSTPILOT
N. LOCO	STSK	OSTSK	XSTSK
N. LOCO	STUDTA	OSTUDTA	XSTUDTA
N. LOCO	SUBMOD	OSUBMOD	XSUBMOD
N. LOCO	SUCOV	OSUCOV	XSUCOV
N. LOCO	SUCPH	OSUCPH	XSUCPH
N. LOCO	SUGYRO	OSUGYRO	XSUGYRO
N. LOCO	SURPLY	OSURPLY	XSURPLY
N. LOCO	SUSHAN	OSUSHAN	XSUSHAN
N. LOCO	SUSTRM	OSUSTRM	XSUSTRM
N. LOCO	SWIND	OSWIND	XSWIND
N. LOCO	SYSINIT	OSYSIN	XSYSIN

```
N. LOCO   TGT50    OTGT50   XTGT50
N. LOCO   THINKPI  OTHINKP  XTHINKP
N. LOCO   TIMER    OTMER    XTIMER
N. LOCO   TOWER    OTOWER   XTOWER
N. LOCO   TRN      OTRN     XTRN
N. LOCO   TURN     OTURN    XTURN
N. LOCO   VARIMOD  OVARIMO  XVARIMOD
N. LOCO   VOICTST  OVOICTS  XVOICTS
N. LOCO   VSPRES   OVSPRES  XVSPRES
N. LOCO   WALOFF   OWALOFF  XWALOFF
N. LOCO   WAVE     OWAVE    XWAVE
N. LOCO   WHEELS   OWHEELS  XWHEELS
N. LOCO   WIND     OWIND    XWIND
N. LOCO   WRMES    OWRMES   XWRMES
N. LOCO   YORN     OYORN    XYORN

. END
```

```
;PART 3 OF LOAD ON CALL TABLE
;---------------------------

N.LOCO  AFAPGP   OAFAP    XAFAP
N.LOCO  AFDNA    OAFDNA   XAFDNA
N.LOCO  AFWC     OAFWC    XAFWC
N.LOCO  CK120    OCK120   XCK120
N.LOCO  CKACK    OCKACK   XCKACK
N.LOCO  CKADH    OCKADH   XCKADH
N.LOCO  CKAGP    OCKAGP   XCKAGP
N.LOCO  CKBD     OCKBD    XCKBD
N.LOCO  CKCHK    OCKCH    XCKCH
N.LOCO  CKCLR    OCKCL    XCKCL
N.LOCO  CKCN     OCKCN    XCKCN
N.LOCO  CKCOR    OCKCOR   XCKCOR
N.LOCO  CKCWO    OCKCWO   XCKCWO
N.LOCO  CKEZN    OCKEZ    XCKEZ
N.LOCO  CKFCP    OCKFCP   XCKFCP
N.LOCO  CKGMR    OCKGMR   XCKGMR
N.LOCO  CKHDCOR  OCKHD    XCKHD
N.LOCO  CKHN     OCKHN    XCKHN
N.LOCO  CKHO     OCKHO    XCKHO
N.LOCO  CKICS    OCKIC    XCKIC
N.LOCO  CKK3     OCKK3    XCKK3
N.LOCO  CKK5     OCKK5    XCKK5
N.LOCO  CKLAA    OCKLA    XCKLA
N.LOCO  CKNGA    OCKNG    XCKNG
N.LOCO  CKOLT    OCKOL    XCKOL
N.LOCO  CKOVR    OCKOV    XCKOV
N.LOCO  CKP18    OCKP18   XCKP18
N.LOCO  CKPAT    OCKPA    XCKPA
N.LOCO  CKPCLR   OCKPC    XCKPC
N.LOCO  CKRFR    OCKRF    XCKRF
N.LOCO  CKRNG    OCKRNG   XCKRNG
N.LOCO  CKROM    OCKROM   XCKROM
N.LOCO  CKTB     OCKTB    XCKTB
N.LOCO  CKTLS    OCKTL    XCKTL
N.LOCO  CKWO     OCKWO    XCKWO
N.LOCO  CKZN3    OCKZN    XCKZN
N.LOCO  DHCK     ODHCK    XDHCK
N.LOCO  HOCK     OHOCK    XHOCK
N.LOCO  OLTCK    OOLTCK   XOLTCK
N.LOCO  P01A     OP01A    XP01A
N.LOCO  P01B     OP01B    XP01B
N.LOCO  P01C     OP01C    XP01C
N.LOCO  P01D     OP01D    XP01D
N.LOCO  P02A     OP02A    XP02A
N.LOCO  P02B     OP02B    XP02B
N.LOCO  P02C     OP02C    XP02C
N.LOCO  P03      OP03     XP03
```

N. LOCO	P04A	OP04A	XP04A
N. LOCO	P04B	OP04B	XP04B
N. LOCO	P04C	OP04C	XP04C
N. LOCO	P04D	OP04D	XP04D
N. LOCO	P05	OP05	XP05
N. LOCO	P05SCH	OP05S	XP05S
N. LOCO	P06	OP06	XP06
N. LOCO	P07A	OP07A	XP07A
N. LOCO	P07B	OP07B	XP07B
N. LOCO	P07C	OP07C	XP07C
N. LOCO	P08	OP08	XP08
N. LOCO	P09A	OP09A	XP09A
N. LOCO	P09B	OP09B	XP09B
N. LOCO	P10A	OP10A	XP10A
N. LOCO	P10B	OP10B	XP10B
N. LOCO	P10C	OP10C	XP10C
N. LOCO	P10D	OP10D	XP10D
N. LOCO	P11A	OP11A	XP11A
N. LOCO	P12A	OP12A	XP12A
N. LOCO	P12B	OP12B	XP12B
N. LOCO	P12C	OP12C	XP12C
N. LOCO	P13A	OP13A	XP13A
N. LOCO	P13B	OP13B	XP13B
N. LOCO	P13C	OP13C	XP13C
N. LOCO	P14A	OP14A	XP14A
N. LOCO	P14B	OP14B	XP14B
N. LOCO	P14SCH	OP14S	XP14S
N. LOCO	P15A	OP15A	XP15A
N. LOCO	P15BC	OP15B	XP15B
N. LOCO	P15SCH	OP15S	XP15S
N. LOCO	P16	OP16	XP16
N. LOCO	P17A	OP17A	XP17A
N. LOCO	P17B	OP17B	XP17B
N. LOCO	P17SCH	OP17S	XP17S
N. LOCO	P18	OP18	XP18
N. LOCO	PATCK	OPATC	XPATC
N. LOCO	PEXCAM	OPEXCAM	XPEXCAM
N. LOCO	PHOSCH	OPHOSC	XPHOSC
N. LOCO	PMCAM	OPMCA	XPMCA
N. LOCO	PMOLT	OPMOL	XPMOL
N. LOCO	PMWAV	OPMWA	XPMWA
N. LOCO	PPANEL	OPPAN	XPPAN
N. LOCO	PSPCH	OPSPC	XPSPC
N. LOCO	PSPEC	OPSPE	XPSPE
N. LOCO	PST1	OPST1	XPST1
N. LOCO	PTURN	OPTURN	XPTURN
N. LOCO	PWAVE	OPWAVE	XPWAVE
N. LOCO	WOCK	OWOCK	XWOCK

CPU 2 BACKGROUND CROSS-REFERENCE TABLE

N. LOCO	COLLECT	OCOLLEC	XCOLLEC
N. LOCO	FORMIT	OFORMIT	XFORMIT
N. LOCO	HELLO	OHELLO	XHELLO
N. LOCO	INIT2R	OINIT2R	XINIT2R
N. LOCO	LEVEL	OLEVEL	XLEVEL
N. LOCO	PLATEXT	OPLATEX	XPLATEXT
N. LOCO	PRESEN	OPRESEN	XPRESENT
N. LOCO	SFORMIT	OSFORMI	XSFORMI
N. LOCO	SKBRD	OSKBRD	XSKBRD
N. LOCO	SKPRO	OSKPRO	XSKPRO
N. LOCO	STIFLE	OSTIFLE	XSTIFLE
N. LOCO	STOVER	OSTOVER	XSTOVER
N. LOCO	STUDST	OSTUDST	XSTUDSTATS
N. LOCO	SUSON	OSUSON	XSUSON
N. LOCO	TEST	OTEST	XTEST
N. LOCO	TUNIT	OTUNIT	XTUNIT
N. LOCO	VALYZ	OVALYZ	XVALYZ
N. LOCO	VCHOS	OVCHOS	XVCHOS
N. LOCO	VCOMP	OVCOMP	XVCOMP
N. LOCO	VDCON	OVDCON	XVDCON
N. LOCO	VDC1VAL	OVDC1VA	XVDC1VA
N. LOCO	VDC2VAL	OVDC2VA	XVDC2VA
N. LOCO	VRPRT	OVRPRT	XVRPRT
N. LOCO	VSPCL	OVSPCL	XVSPCL

APPENDIX M

ERROR EXPLANATIONS

Performance Measurement Variable[1]	Bit[1]	Other Feed-back[2]	Error Explanation
PV00	0	-1,-1,	The pattern controller was not monitored
	1	-1,-1,	"Approaching glidepath" was not said
	2	-1,-1,	"Do not acknowledge" was not said
	3	-1,-1,	"Begin descent" was not said
	4	-1,-1,	"At decision height" was not said
	5	-1,-1,	Clearance was not requested
	6	-1,-1,	Clearance or waveoff was not given
	7	-1,-1,	"Over landing threshold" was not said
	8	-1,-1,	Rollout instructions were not given
	9	-1,-1,	Handoff to pattern controller was not done
	10	-1,-1,	The frequency was not released after handoff
	11	-1,-1,	"No-gyro approach" not announced
	12	-1,-1,	"Make half standard rate turns" not announced at correct time
PV01	2	-1,-1,	The handoff frequency was not monitored
		-1,-1,	The frequency specified in handoff must be monitored
		-1,-1,	
	3	-1,-1,	The handoff was not acknowledged prior to radar contact
		-1,-1,	The handoff must be acknowledged prior to radar contact
		-1,-1,	The pattern controller needs to know if you understood the message
	4	-1,-1,	Handoff not acknowledged within 10 secs of issuance
		-1,-1,	Handoff must be acknowledged within 10 secs
		-1,-1,	The pattern controller needs to know if you understood the message
	5	-1,-1,	Radar contact not reported prior to radio check
		-1,-1,	Radar contact must be reported prior to radio check
		-1,-1,	Pattern controller will not relinquish frequency until radar contact reported
	6	-1,-1,	50% of target not on display when radar contact reported
		-1,-1,	At least 50% of target should be on display when radar contact reported
		-1,-1,	
	7	-1,-1,	Radar contact not reported within 10 secs of 50% of target appearing on display
		-1,-1,	Must report radar contact within 10 secs of 50% of target appearing on display
		-1,-1,	Pattern controller will not relinquish frequency until radar contact reported
	8	0, 2,	The correct call sign not used with radar contact
		0, 2,	The correct call sign must be used with radar contact
		-1,-1,	
	9	0, 1,	Correct frequency not used with radar contact
		0, 1,	Use correct frequency with radar contact
		-1,-1,	
	10	-1,-1,	"Give me..." not said, in order to get unreleased frequency
		-1,-1,	"Give me..." must be said within 15 secs to get unreleased frequency
		-1,-1,	If you do not have the frequency, you cannot control the aircraft
	11	0,-1,	"Give me..." said after pattern controller releases frequency
		0,-1,	Do not say "Give me..." after pattern controller releases frequency
		-1,-1,	
	12	-1,-1,	ICS not deselected when pattern controller released frequency
		-1,-1,	You must deselect ICS when the pattern controller releases the frequency
		-1,-1,	The pattern controller does not want to hear your communication with the pilot

Performance Measurement Variable[1]	Bit[1]	Other Feed-back[2] Error Explanation
PV02	1	-1,-1,Radio contact not established within 30 secs after 50% of target appears
		-1,-1,Establish radio contact within 30 secs after 50% of target appears on display
		-1,-1,If you do not establish contact, you cannot control the aircraft
	2	-1, 1,Proper frequency not selected for radio contact
		-1, 1,Select correct frequency for radio contact
		-1,-1,If you do not select the correct frequency, you cannot communicate with pilot
	3	-1,-1,Mike not keyed when radio contact attempted
		-1,-1,Key mike when establishing radio contact
		-1,-1,The pilot cannot hear you if you do not key the mike
	4	0, 2,Correct call sign not used with radio contact
		0, 2,Use correct call sign with radio contact
		-1,-1,
	5	0,-2,Incorrect phrase used for radio contact
		0,-2,Usé correct phrase for radio contact
		-1,-1,Choose from: "How do ...," "Wheels should...," "Turn ...heading," and "Turn..."
	6	-1,-1,Mike not unkeyed within 3 secs and/or mike not left unkeyed for 5 secs
		-1,-1,Unkey mike within 3 secs and leave unkeyed for 5 secs
		-1,-1,You must give the pilot a chance to respond to radio check
	7	-1,-1,Speech level was not adequate
		-1,-1,Position mike properly and speak clearly
		-1,-1,If your speech is not adequate the pilot cannot follow your instructions
	8	-1,-1,Speech level remained inadequate
		-1,-1,You should have said, "How do you here me now?"
		-1,-1,The pilot will not understand you if your speech level is inadequate
	9	-1,-1,Speech level was not adequate
		-1,-1,Position mike properly and speak clearly
		-1,-1,If your speech is not adequate the pilot cannot follow your instructions
PV03	1	-1,-1,Target not within 2 target widths of cursor at 6 miles
		-1,-1,Target should be within 2 target widths of cursor at 6 miles
		-1,-1,
	2	-1,-1,Target not intercepting azimuth cursor in zones 1 or 2 at 5 miles
		-1,-1,Target should intercept azimuth cursor in zones 1 or 2 at 5 miles
		-1,-1,
	3	-1,-1,Only 1 turn used on turn-to-final
		-1,-1,Use more than one turn for turn-to-final
		-1,-1,Using 1 turn leads to S-turning
	7	0,-1,Turn was in wrong direction
		0,-1,Make turns in proper direction
		-1,-1,
	9	0, 2,Correct call sign not given on turn-to-final
		0, 2,Use correct call sign with turn-to-final
		-1,-1,Without the call sign the pilot doesn't know he is being addressed
	10	0,-1,Turn was in wrong direction
		0,-1,Make turns in proper direction
		-1,-1,
	12	0, 2,Correct call sign not given on turn-to-final
		0, 2,Use correct call sign with turn-to-final
		-1,-1,Without the callsign the pilot doesn't know he is being addressed
	13	0,-1,Turn was in wrong direction
		0,-1,Make turns in proper direction
		-1,-1,
	15	0, 2,Correct call sign not given on turn-to-final
		0, 2,Use correct call sign with turn-to-final
		-1,-1,Without the call sign the pilot doesn't know he is being addressed

Performance Measurement Variable[1]	Bit[1]	Other Feed-back[2] Error Explanation
PV04	1	-1,-1,Call sign and "over" used after "Do not acknowledge" issued
		-1,-1,Do not use call sign and "over" after "Do not acknowledge" issued
		-1,-1,
	2	0, 2,Correct call sign or "over" not used before "Do not acknowledge" issued
		0, 2,Use corrct call sign and "over" before "Do not acknowledge" is issued
		-1,-1,
	3	3, 4,"Approaching glidepath" not given in correct range
		3, 4,Issue "approaching glidepath" in correct range
		-1,-1,This advisory tells the pilot that he is at a particular range
	4	0,-1,"Approaching glidepath" given more than once
		0,-1,Issue "aproaching glidepath" only once during approach
		-1,-1,
	6	0,-1,Correct call sign not used with "Do not acknowledge..." advisory
		0,-1,Correct call sign should be used with "Do not acknowledge..." advisory
		-1,-1,
	7	0,-1,"Over" used with "Do not acknowledge..." advisory
		0,-1,Do not use "over" with "Do not acknowledge..." advisory
		-1,-1,"Over" is a request for response which conflicts with the advisory
	8	-1,-1,"Do not acknowledge..." not given before "begin descent"
		-1,-1,"Do not acknowledge..." should be issued befor "Begin descent"
		-1,-1,
	10	-1,-1,"Begin descent" not transmitted 10-30 secs after "Approaching glidepath"
		-1,-1,Issue "Begin descent" 10-30 secs after "Approaching glidepath"
		-1,-1,
	11	0,-1,Elevation cursor not intersecting top 1/3 of target when "Begin descent" given
		0,-1,Elevation cursor should intersect top 1/3 of target when "Begin descent" given
		-1,-1,
	12	0,-1,"Begin descent" issued more than once
		0,-1,Do not issue "Begin descent" more than once
		-1,-1,
	13	-1,-1,Wheel check not given before "Approaching G/P," pilot has not said "wheels down"
		-1,-1,Wheel check must be given before "Approaching glidepath" given
		-1,-1,
	14	0,-1,Wheel check given after pilot said "Wheels down..."
		0,-1,Do not issue wheel check after pilot says "Wheels down..."
		-1,-1,"Wheels down..." makes the wheel check unnecessary
	15	0,-1,Correct call sign and/or "over" not used with wheel check
		0,-1,Use correct call sign and "over" with wheel check
		-1,-1,
PV05	1	0,-1,Turn not divisible by 5, while range greater than 5 miles
		0,-1,All turns must be evenly divisible by 5, while the range is greater than 5 miles
		-1,-1,
	2	0, 5,You gave a turn of 1 degree
		0, 5,Turns must not be of 1 degree
		-1,-1,
	3	0,-1,360 degree turn given
		0,-1,Do not give 360 degree turns
		-1,-1,If the pilot responds to the direction and not the heading, he will turn wrong
	4	-1,-1,120 degree turn issued without a counter-corrective turn within 8 secs
		-1,-1,Issue a counter-corrective turn within 8 secs of a 120 degree turn
		-1,-1,
	5	-1,-1,Target entered zone 3 from 2 without corrective turn within 20 secs
		-1,-1,Issue a corrective turn within 20 secs of target entering zone 3 from 2
		-1,-1,
	7	0,-1,Heading given in a "Heading..." advisory not the same as previous assignment
		0,-1,Heading used in a "Heading..." advisory must be the same as previous assignment
		-1,-1,
	8	0,-1,"Heading..." used more than 5 times during an approach
		0,-1,"Heading..." must not be used more than 5 times during an approach
		-1,-1,

Performance Measurement Variable[1]	Bit[1]	Other Feed-back[2] Error Explanation
PV06	1	0, 6,Incorrect position given in azimuth position call
		0, 6,Give correct position in azimuth position calls
		-1,-1,
	2	-1,-1,"Correcting" not given within 3 secs of "well..." advisory, target closing
		-1,-1,Give "correcting" advisory within 3 secs of "well" if aircraft is closing
		-1,-1,
	3	-1,-1,Corrective turn not given within 3 secs of a "well...", target not closing
		-1,-1,Give a corrective turn within 3 secs of a "well..." , if target is not closing
		-1,-1,
	4	0,-1,"Correcting" used when target not closing with centerline
		0,-1,Do not use "correcting" if target is not closing with centerline
		-1,-1,
PV07	1	0,-1,Glidepath message given before "Begin descent"
		0,-1,Give "Begin descent" before any glidepath messages
		-1,-1,
	2	0, 7,Incorrect position in glidepath position call
		0, 7,Use correct position in glidepath position call
		-1,-1,
	3	0,-1,Target changed zones without a position call
		0,-1,Issue a position call whenever target changes zones
		-1,-1,
	4	0, 8,Incorrect trend call
		0, 8,Issue correct trend calls
		-1,-1,
	5	0,-1,Trend message not given after target moves from one zone to another
		0,-1,Trend message must be given if aircraft moves from one zone to another
		-1,-1,
	6	0,-1,Trend messages issued successively inside of well zone
		0,-1,Do not issue trend messages successively unless aircraft is in well zone
		-1,-1,
	7	0,-1,Identical position calls separated by trend message outside of well zone
		0,-1,Trend messags must not separate identical position calls except in well zone
		-1,-1,
PV08	11	-1,-1,Range call omitted
		-1,-1,All range calls must be made after one is made or 5 miles, unless superseded
		-1,-1,
	12	0,-1,Range call not given within .1 mile of range mark
		0,-1,Range calls must be made within .1 mile of range mark
		-1,-1,
	13	0, 4,Incorrect range used in range call
		0, 4,Use correct range in range calls
		-1,-1,
PV09	2	0, 9,Highest priority message not given or incorrect position at decision height
		0, 9,Highest priority correct position must be given at decision height
		-1,-1,Pilot must know most important position error
	3	-1,-1,The highest priority call was "Too low"
		-1,-1,The highest priority call was "Too low"
		-1,-1,
	4	-1,-1,The highest priority call was "Too far left"
		-1,-1,The highest priority call was "Too far left"
		-1,-1,
	5	-1,-1,The highest priority call was "Too far right"
		-1,-1,The highest priority call was "Too far right"
		-1,-1,

Performance Measurement Variable[1]	Bit[1]	Other Feed-back[2] Error Explanation
PV09 Cont.	6	-1,-1,The highest priority call was "Too high"
		-1,-1,The highest priority call was "Too high"
		-1,-1,
	7	0,16,"At decision height" announced prior to .8 miles from touchdown
		0,16,"At decision height" must be announced prior to .8 miles from touchdown
		-1,-1,
	8	0,16,"At decision height" not announced prior to .7 miles from touchdown
		0,16,"At decision height" must be announced prior to .7 miles from touchdown
		-1,-1,This point has been carefully selected to provide optimum safety
	9	0,-1,"At decision height" announced twice
		0,-1,Announce "At decision height" only once during approach
		-1,-1,
PV10	1	-1, 4,Initial clearance request issued prior to 3.1 miles from touchdown
		-1, 4,Do not request clearance prior to 3.1 miles from touchdown
		-1,-1,
	2	-1,-1,Initial clearance request not made prior to 2.9 miles from touchdown
		-1,-1,Make initial clearance request prior to 2.9 miles from touchdown
		-1,-1,
	3	-1,-1,Second clearance request not made between 2.1 and 1.9 miles from touchdown
		-1,-1,If clearance not received, a second must be made between 2.1 and 1.9 miles
		-1,-1,This range has been calculated to provide for a safe waveoff, if necessary
	4	-1,-1,Second clearance request made after clearance issued from tower
		-1,-1,If clearance is received, a second request should not be made
		-1,-1,
	5	0,10,Correct wind information not issued
		0,10,Correct wind information should be issued
		-1,-1,
	6	0,-1,Wind information issued before clearance received from tower
		0,-1,Wind information should not be issued to pilot until received from tower
		-1,-1,Many pilots take wind iformation to mean that clearance has been received
	7	0,16,Clearance issued to pilot before received from tower
		0,16,Do not issue clearance to pilot before received from tower
		-1,-1,
	8	0,-1,Wind information not issued with clearance
		0,-1,Wind information must be issued just before clearance
		-1,-1,
	9	-1,-1,Clearance not issued prior to 1 mile
		-1,-1,Issue clearance prior to 1 mile
		-1,-1,
	10	16,-2,Reason and waveoff not issued prior to 1.3 miles
		16,-2,Issue reason and waveoff prior to 1.3 miles
		-1,-1,You must give Tower clearance not received and execute missed approach ...
	11	0,11,Improper missed approach advisory with waveoff
		0,11,Use proper missed approach advisory with waveoff.
		-1,-1,
	12	16,-2,Reason and waveoff not issued within 2 secs of clearance cancellation
		16,-2,Issue reason and waveoff within 2 secs of clearance cancellation
		-1,-1,You must give Tower clearance cancelled and execute missed approach ...
PV11	2	0,-1,OVER LANDING THRESHOLD not issued within 1 sec of target contacting LT
		0,-1,OVER LANDING THRESHOLD must be issued within 1 sec of target touching LT
		-1,-1,
	3	0,-1,Final course position not given within 3 secs of OLT
		0,-1,Final course position must be given within 3 secs of OLT
		-1,-1,
	4	0, 6,Incorrect final course position
		0, 6,The final course position must be correct
		-1,-1,
	5	0,-1,"Over" not used with final course position
		0,-1,"Over" must be used with final course position
		-1,-1,

Performance Measurement Variable[1]	Bit[1]	Other Feed-back[2] Error Explanation
PV12	2	0,-1,Rollout instructions not issued 20-40 secs after "over" 0,-1,Rollout instructions must be issued 20-40 secs after "over" -1,-1,
	3	-1,-1,The radio frequency was not released within 10 secs of rollout instructions -1,-1,The radio frequency must be released within 10 secs of rollout instructions -1,-1,
	4	-1,-1,Pattern controller not notified after rollout instructions given -1,-1,Notify pattern controller after giving rollout instructions -1,-1,
	6	-1,12,Handoff to pattern controller not given in allotted time -1,12,An ICS report for a missed approach must be made within 30 seconds -1,-1,
	7	0, 2,Correct call sign not given with handoff 0, 2,Correct call sign must be given with handoff -1,-1,
	9	0,13,Correct button not given with handoff 0,13,Give correct button with handoff -1,-1,
	10	0, 4,Range not reported to nearest 1/2 mile on missed approach 0, 4,Range must be reported to nearest 1/2 mile on missed approach -1,-1,
	11	0,-1,Range reported to pattern controller on non-missed approach 0,-1,Do not report range to pattern controller if not a missed approach -1,-1,
	12	-1,-1,Frequency or ICS not monitored until pattern controller said "<Call sign> radar" -1,-1,Monitor frequency and ICS until pattern controller says "<Call Sign> radar" -1,-1,
	13	-1,-1,Radio frequency not released after handoff -1,-1,Release radio frequency after handoff. -1,-1,
	14	-1,-1,Pattern ICS not selected during handoff -1,-1,Pattern ICS must be selected during handoff -1,-1,
PV13	1	-1,-1,"Assigned heading XXX" not announced when turn caused only a 2 deg course change -1,-1,Announce "Assigned heading XXX" if there's only a 2 deg course change -1,-1,
	2	0,-1,No-gyro approach not announced after correction not taken within 1/2 mile 0,-1,No-gyro approach must be announced if course correction not taken within 1/2 mile -1,-1,
	3	0,-1,No-gyro approach not announced at correct time 0,-1,Announce no-gyro approach within 3/4 mi from issuing warning -1,-1,
	5	0,-1,"Make half...turns" announced before "No-gyro approach" or "Begin descent" 0,-1,Do not announce "Make half..." before "No-gyro" or "Begin descent" -1,-1,
	6	0,-1,"Make half standard rate turns" announced more than once 0,-1,"Make half standard rate turns" should be announced only once -1,-1,
PV14	1	0,14,No-gyro turn in wrong direction 0,14,No-gyro turns must be in correct direction -1,-1,
	2	0,-1,"Stop turn" not issued on no-gyro approach 0,-1,"Stop turn" must be issued on no-gyro approach -1,-1,The pilot cannot tell from his instruments when to stop a turn
	3	0,-1,Heading correction not made within 20 secs of target entering zone 3 from 2 0,-1,Make heading correction within 20 secs of target entering zone 3 from zone 2 -1,-1,

Performance Measurement Variable[1]	Bit[1]	Other Feed-back[2] Error Explanation
PV15	1	0,-1,"Radar contact lost" used incorrectly 0,-1,Issue "Radar contact lost" only when target disappears -1,-1,
	2	-1,-1,Waveoff not issued within 5 secs of aircraft moving off screen -1,-1,Issue a waveoff within 5 seconds of aircraft moving off screen -1,-1,
	3	0,15,Improper radio terminology used with a waveoff 0,15,Use proper radio terminology with a waveoff -1,-1,
	10	-1,-1,Improper radio terminology used with a waveoff -1,-1,Use proper radio terminology with a waveoff -1,-1,
	11	16,-1,Waveoff did not follow "Too..." message at decision height 16,-1,Waveoff must follow "Too..." message at decision height -1,-1,
	12	16,-1,"Too..." message not used or improperly used with waveoff at decision height 16,-1,Proper "Too..." message must be used with waveoff at decision height -1,-1,
PV16	2	-1,-1,Low altitude alert not issued within 5 secs of onset of low altitude condition -1,-1,Issue low altitude alert within 5 secs of onset of low altitude condition -1,-1,This occurs when target's distance below cursor exceeds 1 target width/mi from TD
PV17	1	-1,-1,Mike was not unkeyed after "over" -1,-1,Unkey mike after "over" -1,-1,This gives the pilot a chance to give information
	2	-1,-1,Mike not unkeyed at least once between "Do not acknowledge..." and 1 mile -1,-1,Mike must be unkeyed at least once between "Do not acknowledge" and 1 mile -1,-1,This gives the pilot a chance to offer information
PV18	2	-1,-1,There was more than 5 secs between advisories after "Do not acknowledge..." -1,-1,There should be no more than 5 secs between advisories after DNA -1,-1,A break of more than 5 secs will cause a missed approach

Performance Measurement Variable[1]	Bit[1]	Other Feed-back[2]	Error Explanation
PV19	1	-1,-1,	Azimuth alignment needed.
		-1,-1,	Azimuth alignment needed.
		-1,-1,	
	2	-1,-1,	Azimuth alignment not needed.
		-1,-1,	Azimuth alignment not needed.
		-1,-1,	
	3	-1,-1,	Elevation alignment needed.
		-1,-1,	Elevation alignment needed.
		-1,-1,	
	4	-1,-1,	Elevation alignment not needed.
		-1,-1,	Elevation alignment not necessary.
		-1,-1,	
	5	-1,-1,	Range alignment needed.
		-1,-1,	Range alignment necessary.
		-1,-1,	
	6	-1,-1,	Range alignment not needed.
		-1,-1,	Range alignment not necessary.
		-1,-1,	
	7	-1,-1,	Azimuth not servoed down before checking azimuth and range alignment.
		-1,-1,	Before aligning azimuth and range, servo down until centerline reflector appears
		-1,-1,	
	8	-1,-1,	Elevation radar not servoed left correctly before checking elevation alignment
		-1,-1,	Before checking elevation alignment, servo left 'til touchdown reflector appears
		-1,-1,	
	9	-1,-1,	Azimuth not servoed up correctly after alignment procedures
		-1,-1,	After checking alignment, servo up until 1 mile mark is bisected by glideslope
		-1,-1,	
	10	-1,-1,	Elevation not servoed correctly after alignment procedure
		-1,-1,	After checking alignment, servo so azimuth cursor is bisected by 1 mile mark
		-1,-1,	

NOTES

1 See Tables 27 - 46.

2 See description of the contents of words 1 and 2 of the Error Explanation File in Appendix C.